**A Computational Tool for the Rapid Design and Prototyping
of Propellers for Underwater Vehicles**

by

Kathryn Port D'Epagnier

B.S. Ocean Engineering
The United States Naval Academy (2005)

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mechanical Engineering
at the
Massachusetts Institute of Technology
and the
Woods Hole Oceanographic Institution

September 2007

Signature of Author_____
Department of Mechanical Engineering
Aug 24, 2007

Certified by_____
Patrick J. Keenan
Professor of the Practice of Naval Architecture
Thesis Supervisor

Certified by_____
Richard W. Kimball
Thesis Supervisor

Accepted by_____
Lallit Anand
Professor of Mechanical Engineering
Chairman, Departmental Committee on Graduate Students

1. REPORT DATE (DD-MM-YYYY) XX-06-2007	2. REPORT TYPE Master's Thesis	3. DATES COVERED (From - To) JAN-JUN 2007

4. TITLE AND SUBTITLE A Computational Tool for the Rapid Design and Prototyping of Propellers for Underwater Vehicles	5a. CONTRACT NUMBER N62271-97-G-0026
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Kathryn Port D'Epagnier	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute of Technology	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, Ca 93943	10. SPONSOR/MONITOR'S ACRONYM(S) NPS
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
1. DISTRIBUTI9ON STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
An open source, MATLAB™-based propeller design code MPVL was improved to include rapid prototyping capabilities as well as other upgrades as part of this effort. The resulting code, OpenPVL is described in this thesis. In addition, results from the development code BasicPVL are presented. An intermediate code, BasicPVL, was created by the author while OpenPVL was under development, and it provides guidance for initial propeller designs and propeller efficiency analysis. OpenPVL is part of the open source software suite of propeller design codes, OpenProp. OpenPVL is in the form of a Graphical User Interface (GUI) which features both a parametric design technique and a single propeller geometry generator. This code combines a user-friendly interface with a highly modifiable platform for advanced users. This tool offers graphical propeller design feedback while recording propeller input, output, geometry, and performance. OpenPVL features the ability to translate the propeller design geometry into a file readable by a Computer Aided Design (CAD) program and converted into a 3D-printable file. Efficient propellers reduce the overall power requirements for Autonomous Underwater Vehicles (AUVs), and other propulsion-powered vehicles. The focus of this study is based on the need of propeller users to have an open source computer-based engineering tool for the rapid design of propellers suited to a wide range of underwater vehicles. Propeller vortex lattice lifting line (PVL) code in combination with 2D foil theory optimizes propeller design for AUVs. Several case studies demonstrate the functionality of OpenPVL, and serve as guides for future propeller designs. The first study analyzes propeller thruster performance characteristics for an off-the-shelf propeller, while the second study demonstrates the process for propeller optimization—from the initial design to the final file that can be read by a 3D printer. The third study reviews the complete process of the design and production of an AUV propeller. Thus, OpenPVL performs a variety of operations as a propeller lifting line code in streamlining the propeller optimization and prototyping process.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Sean Tibbitts, Educational Technician
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	95	19b. TELEPHONE NUMBER (include area code) (831) 656-2319 civins@nps.edu

This page intentionally left blank.

A Computational Tool for the Rapid Design and Prototyping
of Propellers for Underwater Vehicles
by

Kathryn Port D'Epagnier

Submitted to the Department of Mechanical Engineering on August 24, 2007
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Abstract

An open source, MATLAB™-based propeller design code MPVL was improved to include rapid prototyping capabilities as well as other upgrades as part of this effort. The resulting code, *OpenPVL* is described in this thesis. In addition, results from the development code BasicPVL are presented. An intermediate code, BasicPVL, was created by the author while OpenPVL was under development, and it provides guidance for initial propeller designs and propeller efficiency analysis. OpenPVL is part of the open source software suite of propeller design codes, OpenProp. OpenPVL is in the form of a Graphical User Interface (GUI) which features both a parametric design technique and a single propeller geometry generator. This code combines a user-friendly interface with a highly modifiable platform for advanced users. This tool offers graphical propeller design feedback while recording propeller input, output, geometry, and performance. OpenPVL features the ability to translate the propeller design geometry into a file readable by a Computer Aided Design (CAD) program and converted into a 3D-printable file.

Efficient propellers reduce the overall power requirements for Autonomous Underwater Vehicles (AUVs), and other propulsion-powered vehicles. The focus of this study is based on the need of propeller users to have an open source computer-based engineering tool for the rapid design of propellers suited to a wide range of underwater vehicles. Propeller vortex lattice lifting line (PVL) code in combination with 2D foil theory optimizes propeller design for AUVs. Several case studies demonstrate the functionality of OpenPVL, and serve as guides for future propeller designs. The first study analyzes propeller thruster performance characteristics for an off-the-shelf propeller, while the second study demonstrates the process for propeller optimization—from the initial design to the final file that can be read by a 3D printer. The third study reviews the complete process of the design and production of an AUV propeller. Thus, OpenPVL performs a variety of operations as a propeller lifting line code in streamlining the propeller optimization and prototyping process.

Thesis Supervisor: Patrick J. Keenan

Title: Professor of the Practice of Naval Architecture

Acknowledgements

The author greatly appreciates the dedicated guidance of Richard W. Kimball, and is thankful for the time he has devoted to sharing his wealth of knowledge on the subject matter in this thesis.

The author expresses thanks to Rear Admiral David A. Gove, Oceanographer of the United States Navy, for academic funding and making this study possible. The author also would also like to thank CAPT P. Keenan, USN, for his support.

Additionally, the author would like to credit the Deep Submergence Lab at Woods Hole Oceanographic Institute, for use of their vehicles and resources in testing for this project.

Furthermore, the author is grateful for the other Navy and MIT/WHOI Joint Program students and the time spent studying and collaborating together. The author thanks the members of her propeller design group, for their contributions, and especially Hsin-Lung Chung for developing the foundations of the open source project OpenProp. The author would also like to thank Jordan Stanway, for providing information on automating RHINO™ scripting.

Finally, the author expresses deep gratitude towards her family and close friends for encouraging both her research and her US Navy career.

Table of Contents

List of Figures

List of Tables

General Nomenclature

c	section chord length	t_o/c	thickness ratio
c/D	ratio of chordlength to diameter	u_a	induced axial velocity
C_D	drag coefficient	u_t	induced tangential velocity
C_L	lift coefficient	V_a/V_s	axial inflow velocity ratio (is 1.0, assuming uniform inflow)
C_{LCHART}	lift coef. From NACA a = 0.8 data (C_L = 1.0)	V_s	ship (vehicle) velocity
C_p	propeller power coefficient	V_t/V_s	tangential inflow velocity ratio (is 0.0, assuming uniform inflow)
C_{Pmin}	local pressure distribution coefficient	V^*	total inflow velocity
C_q	required torque	Z	number of blades
C_T	thrust loading coefficient		
D	Diameter		
$\dfrac{f_o}{c}$	camber ratio	$\Delta\alpha$	Angle of attack variation due to non-uniform flow
f_{oCHART}	max camber ratio from NACA a = 0.8 data ($f_o = 0.679$)	β	undisturbed flow angle (relative to blade)
G	non-dimensional circulation, $G = \dfrac{\Gamma}{\pi R V_s}$	β_i	hydrodynamic pitch angle
H	underwater vehicle operation depth	Γ	vortex circulation strength
J	advance coefficient, $J = V_s/ND$	η	Efficiency
K_q	torque coefficient	ρ	Water density
K_T	propeller thrust coefficient, $K_T = C_T J^2 \pi/8$	σ	cavitation number, $\sigma = \dfrac{Patm + \rho gh - Pvap}{\frac{1}{2}\rho V^{*2}}$
N	propeller Revolutions Per Second		
P_{atm}	atmospheric pressure		
P_{vap}	vapor pressure		
P/D	ratio of pitch over diameter		
r/R	ratio of the radial location, r, to the total length of the blade radius, R		

1 Introduction and Background

1.1 Motivation

Autonomous Underwater Vehicles (AUVs) serve in scientific research and in military operations. Engineers need an adaptable tool that can streamline the propeller design process for a wide range of vehicles, including AUVs. REMUS AUVs have experienced success through use of their own propeller designs, yet such design capabilities need to be made available in the form of open source code, such as OpenProp [1]. In some cases, off-the-shelf propellers are used for underwater vehicles. Off-the-shelf designs are easier to obtain than a custom-designed propeller, yet they are not optimized for the capabilities of a specific vehicle. It is important to consider that a wide variety of underwater vehicles exist. Unique vehicle characteristics call for a tool that has the capacity to design propellers for a specific application.

Underwater propellers are sometimes not optimized due to factors of cost and availability. In designing a vehicle, off-the-shelf components can be very desirable. Traditionally, it would be much more costly, difficult, and time-consuming to design and fabricate a propeller than it would be to order one that has already been tested and can easily be replaced. Some AUV designs, including the SeaBED and ABE varieties of AUVs developed in the Deep Submergence Lab at the Woods Hole Oceanographic Institute (WHOI), incorporate a commercially available model airplane propeller. Manufactured propellers can be appropriately analyzed and chosen by inputting various off-the-shelf blade geometries into OpenPVL and then evaluating the commercial options to determine the propeller that is best fit to serve a vehicle in operation. In this way, off-the-shelf propeller users can benefit from the streamlined design-to-prototype process OpenPVL offers users.

In a situation in which fabricating and testing propellers is prohibitively expensive, or it is desirable to use existing propellers, modifications can be made to improve performance in propeller efficiency. This analysis seeks to provide a means of designing and manufacturing propellers for specific AUV operations, while contributing to methods for improving performance using off-the-shelf model airplane propellers.

Use of propellers in aircraft and ships, both military and commercial, commands the development of modern propeller design tools [2]. Today, computers allow a detailed optimization of propellers. Propeller vortex lattice methods developed by J. Kerwin represent the propeller as a vortex lifting line. Efficiency is maximized by computing the optimal radial distribution of loading (circulation). Hub effects are represented with image vortices and the wake is represented as constant diameter helical vortices aligned with the hydrodynamic pitch of the propeller flow. After optimizing an initial design, the propeller performance undergoes two-dimensional analysis, including determining minimum pressure coefficients and local lift coefficients. Rigorous design methods use multiple iterations of many parameters to calculate a highly efficient propeller design [3].

1.2 Recent Propeller Design Efforts

Vortex lifting line codes provide a powerful method for assessing the characteristics of a propeller. In the initial analysis, Propeller Vortex Lifting Line (PVL) code, was used in combination with MATLAB™ m-files [4]. The m-file created to use with PVL, basicPVL.m, was a developmental code used prior to to the May 2007 release of Hsin-Lung Chung's open source MIT Propeller Vortex Lifting Line code, (MPVL) [5]. MPVL can be utilized to perform both a parametric propeller analysis, and a single propeller design.

The most recently developed design tool is OpenPVL, which is an open source propeller design tool that has the capability to manufacture propeller blades from desired design inputs. The OpenProp suite (including OpenPVL) is based on the FORTRAN programs originally developed by Professor J. Kerwin at MIT in 2001 [4], translated into MATLAB™ as an open source MPVL code released by Hsin-Lung Chung in May 2007 [5]. OpenPVL operates using an evolved MPVL code, while expanding upon MPVL's applications, and it has been modified to create scripting for 3D printable files through the CAD interface commercial software RHINO™. OpenPVL creates a user-friendly interface to quickly input design characteristics and output performance data and propeller blade geometry. Source code for OpenPVL.m can be found in Appendix A. OpenPVL files, including the most recent updates, can be obtained along with the OpenProp suite at http://web.mit.edu/openprop/.

2　Lifting Line Propeller Design Codes

2.1　An Introduction to Design Techniques

There are many methods for propeller design [4,6], but in this study, propeller vortex lifting line code (both basicPVL.m and OpenPVL.m) were analyzed using cavitation considerations to meet the AUV specifications [3]. These methods lead to designing an optimal propeller, as well as calculating the efficiency of existing propellers, which determined performance flaws. The resulting analysis produced recommendations for altering the geometry of off-the-shelf propellers to improve overall efficiency, while considering the effects of minimum pressure coefficients and ranges of lift coefficient. Aspects of interest in propeller design included the following: the vehicle's desired thrust, cruise speed, number of propellers, propeller location, and depth of vehicle. The design additionally considered the affects of blade number, rate of propeller rotation, chord length distribution, propeller diameter, and hub size.

This study made several assumptions. First, the propellers were located far enough away from the hulls of the AUV, which left axial velocity, V_s, unaffected. The ratio of axial velocity to design speed was such that $V_a/V_s = 1.0$. Hull wake and boundary layer effects were not considered in the cases studies presented. However, these effects can be accounted for by using basicPVL, which handles radial variation both axial and tangential velocities. Additionally, hull diameter was insignificant due to the location of the thrusters away from the body of the AUV; thruster diameter was small relative to the propeller diameter. Also, the propeller design code did not automatically check for cavitation effects, but was analyzed by hand using Brockett diagrams. Furthermore, the required thrust was determined for the design speed, and calculated per propeller, not as the total required thrust for the vehicle.

The code basicPVL was designed as an intermediate code to perform analysis during the time that the GUI for OpenPVL was being developed. The analysis considered both the optimal propeller design for the given design criteria, as well as the efficiency of the existing propeller. Efficiency (η), is determined by the ratio of required thrust times the wake to the power coefficient. The actuator disk efficiency can be calculated in addition to η. If a value

11

higher than the actuator disk efficiency is found, it can be eliminated as erroneous. The actuator disk efficiency is calculated as in Equation 2.1 [4]:

$$\eta_{ActuatorDisk} = 2 \Big/ (1 + (1 + K_T)^{1/2}) \ .$$

2.1

This calculation serves as an added check to determine if the calculated efficiency (η) is correct. In finding an initial propeller design, PVL.exe was used with input variables, as in [4], including the advance coefficient, J, the desired thrust coefficient, C_T, and the axial flow ratio, V_a/V_s.

A function was created to determine the effects of axial flow for alternative propeller placement or diameter of a body affecting fluid flow entering the propeller [7].[1] The axial flow modeling considers several important factors. The first factor is in the case that the ratio of the propeller blade diameter to the hull diameter is not equal to one. The data was extended to include a case of a ratio of up to 1.8. Additionally, the practical assumption was made that axial flow is constant for all r/R values less than 0.225. Otherwise, the spline would model a value of zero or less at the hub, which is not possible.

Other variables, such as the number of vortex panels over the radius, iterations in the wake alignment, hub image, hub vortex radius/hub radius, number of input radii, hub unloading factor, tip unloading factor, swirl cancellation factor, radial position (r/R), chord to diameter ratio (c/D), and blade section drag (C_D), were kept at acceptable values from example input files to the PVL code. The c/D values were scaled automatically with respect to r/R values using a spline function.

2.2 Blade Design Using basicPVL

Blade selection was determined by parameterization of a range of acceptable vehicle design speeds, propeller diameters, and number of blades. A range of values were chosen for the blade number (z), revolutions per second (N), and the diameter of the propeller (D). Values for z, N and D were plotted versus propeller efficiency, and shown as contour lines on a 3D plot.

---

[1] Data from the experimental Huang Body 1 relating axial flow to radial position was splined such that it returns values for axial flow for any range and any spacing of values for radial position.

The code for basicPVL presents an understanding of principles of propeller design. Chordlength optimization were crudely accomplished by a manual process. The input values were somewhat limited and desired values for the parameters were hand-typed for each case tested. This method of parameterization lead to the choice of an initial propeller design, as shown in Fig. 2-1.
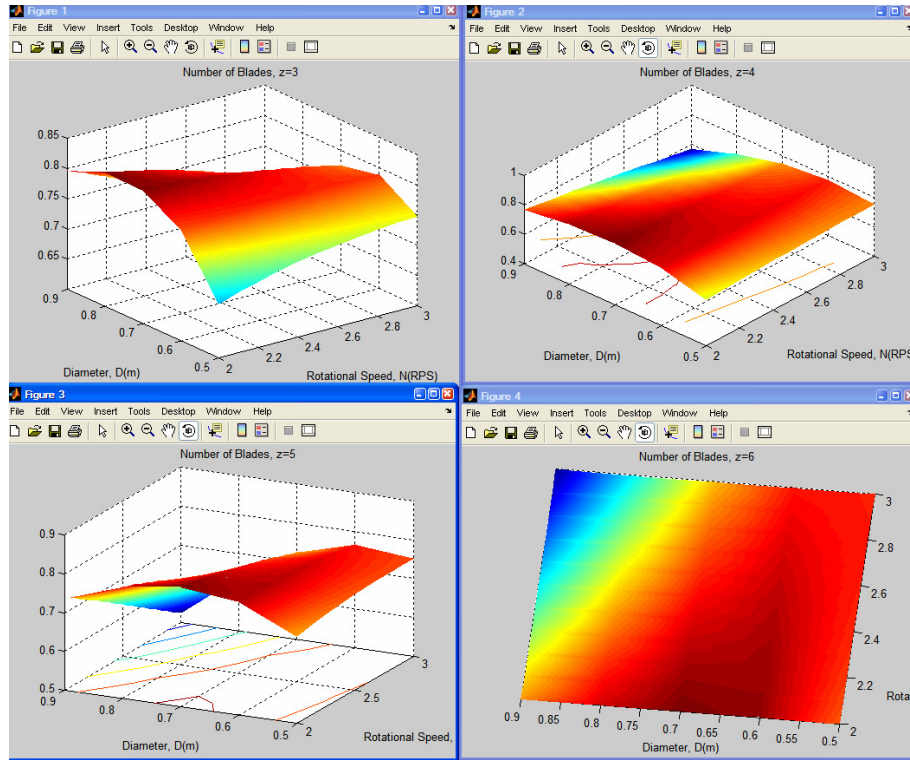
Figure 2-1: Parametric Blade Design using basicPVL.

The intial blade shape parameters, such as chord, section thickness, rake and skew radial distributions are input by the designer. After the intitial run of the code, the chord distribution can be adjusted manually to adjust the section lift and cavitation performance of the blade. The blade section drag coefficient distribution is also user input to account for blade viscous drag effects. For typical NACA foils operating with lift coefficients between 0-0.5 and an assumed uniform blade section drag, C_D=0.008 is used as an estimate in practical applications. Typically, the designer uses a distribution of lift coefficients which is higher at the hub, and tapers to lower at the tip for added strength and stall prevention. Once the shape of the propeller is chosen, the design process continues with an optimization of chordlength. Other

slight parameter adjustments improve overall efficiency. It is in the section solver that pressure distribution and the lift coefficient are considered. The finely adjusted ratio of chordlength over diameter (c/D) depends on an appropriate lift coefficient, C_L. The calculated drag coefficient, C_D, remains relatively constant for lift coefficients from 0.2-0.5, with a higher lift coefficient at the hub, and a lower lift coefficient at the tip of the blade [8]. The lift coefficient at the tip must be lower than at the hub due to blade strength limitations [3]. By remaining within this range of lift coefficients, lift is maximized with minimal sacrifice of efficiency to drag. The lift coefficient can be determined by dimensionalizing gamma, G, at each radial location, using the following two equations (2.1-2.2)[3]:

First, $\Gamma = 2\pi G V_s R$ **2.2**

Then, $C_L = \dfrac{2\Gamma}{V * c}$ **2.3**

The total velocity of the lifting line, V*, can be found by using β_i, and calculating ωr less ut* given from APLOT.plt in PVL. Figure 2.2 shows the representation of the velocity vectors at a particular radius on the lifting line blade. The onflow to the section is V and the induced velocity components ua* and ut* represent the contributions of the propeller, hub, and wake vortex structures to the overall flow. The total flow at the lifting line, including the propeller induction is V*, and the hydrodynamic pitch angle β_i represents the flow angle seen by the blade section used for setting the section camber and angle of attack. The 2D sections are then set at the ideal angle of attack for the chosen section type and the camber is scaled to achieve the computed lift coefficient for the section. In this way, the blade geometry at each radii is determined, resulting in a complete blade geometry.

14

$$V* = \frac{(wr + ut*)}{\cos \mathrm{B}_i}$$

$$\omega r = 2\pi \mathrm{NR(r/R)}$$
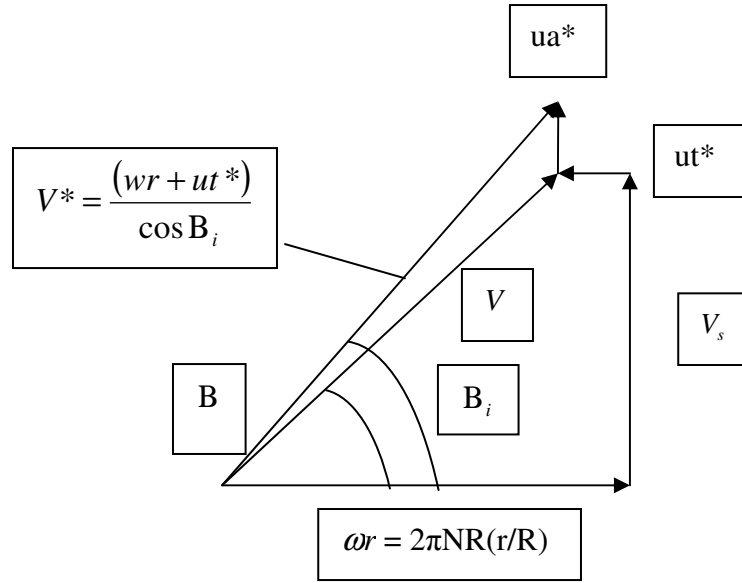
Figure 2-2: Geometry for determining total lifting line velocity, V*, as in [4].

Cavitation is an important design consideration for propellers operating at shallow depth or at high loading. Preliminary cavitation prediction and design is achieved using the procedure of Brockett [9], using the Brocket diagram method for a cavitation-free blade design. At each blade radii, the minimum local operating depths of the section is computed using equation 2.3 and the section cavitation number is computed using equation 2.4. The axial inflow variation is estimated by the designer and converted into an angle of attack variation by applying the inflow variation to the velocity triangle (Fig. 2-2) and calculating the change in inflow angle (β). Next, the Brocket diagrams are used to select the section thickness in order for the foil to achieve cavitation-free performance.   The section chordlength can be increased until the desired performance is achieved. In addition to being less efficient, partial cavitation creates bubbles which can damage propeller blades through pitting [10].   The cavitation number can be compared to the pressure coefficient such that cavitation will not occur if σ > C$_{Pmin}$. The value for the cavitation number, σ, can be found from the following equations (2.3-2.4) [4]:

$$h = h_o - (r/R)(\tfrac{D}{2}), \text{ where } h_o = \text{operating depth.}$$ **2.4**

$$\sigma = \frac{Patm + \rho gh - Pvap}{\tfrac{1}{2}\rho V*^2}.$$ **2.5**

15

The Brockett diagram utilizes the NACA-66 sections [9] in finding the camber ratio, f_o/c, and the inflow variation bucket width, $\Delta\alpha$, is used to determine the thickness ratio, t_o/c, and local pressure distribution coefficient, C_{Pmin} [4].

2.3 Design Techniques Using OpenPVL

OpenPVL is a propeller design tool that is based on the same vortex lattice lifting line theory as basicPVL. However, OpenPVL offers a variety of advanced features, including:

- A user-friendly MATLAB™ GUI
- Ability to save valuable input and output text files for each propeller design
- Capacity to write a script file for 3D printing from the design geometry output

Figure 2-3 shows the main input screen for the OpenPVL code. Two options are available at this level, *Parametric Analysis* and *Single Propeller Design*.
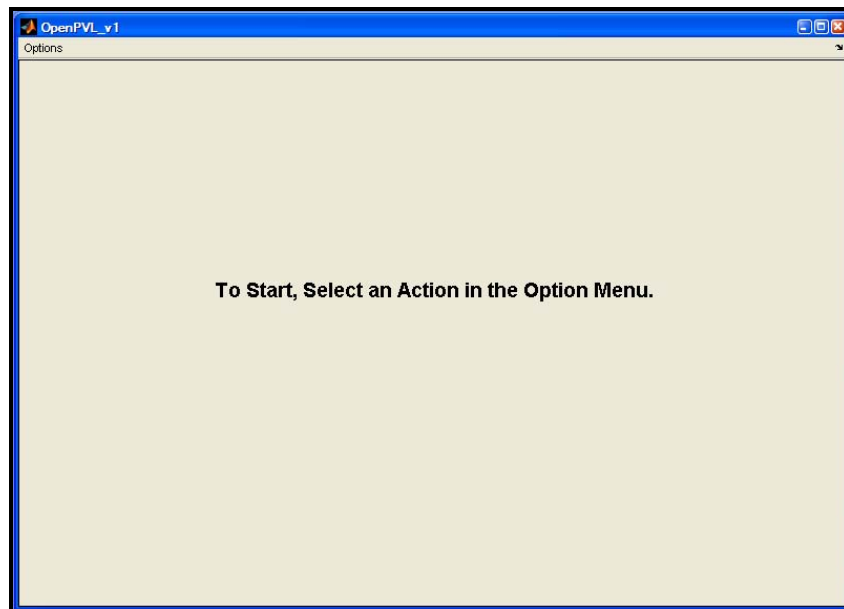
Figure 2-3: Screenshot of the main input screen for OpenPVL code.

16

2.3.1 Parametric Analysis

Three parameters provide the foundation for propeller design: the number of blades, the propeller speed, and the propeller diameter. Various combinations of these three key parameters result in different efficiencies. Thus, a parametric study allows for propeller parameter optimization. The *Parametric Analysis* GUI is a computational tool that calculates and graphically represents propeller efficiency. Figure 2-4 shows the parametric analysis GUI, which includes the user input fields required to run the analysis. OpenPVL is tailored to a propeller user's design needs, therefore, the *Parametric Analysis* GUI requires user input for the following characteristics:

- Number of blades
- Propeller speed
- Propeller diameter
- Required thrust
- Ship speed
- Hub diameter
- Number of vortex panels over the radius
- Maximum number of iterations in wake alignment
- Ratio of hub vortex radius to hub radius
- Number of input radii
- Hub and tip unloading factor
- Swirl cancellation factor
- Water density
- Hub image flag

All of the fields within the GUI are populated with initial values, based on the US Navy 4148 propeller, as a guide to users. Each of the input fields are modifiable and *Parametric Analysis* can run any desired number of times without having to exit the program [5].

Figure 2-4: *Parametric Analysis* **GUI.**

2.3.2 Single Propeller Design

Once the parameters for a propeller with a viable efficiency curve has been established, the desired inputs are entered into the *Single Propeller Design* GUI of OpenPVL. Figure 2-5 shows the single propeller design GUI. Determining the geometry for a single propeller utilizes both the results from the propeller parameterization, as well as additional inputs, resulting in a user-specific design. Input fields entered for the *Parametric Analysis* are populated with the same values for the *Single Propeller Design*. There are also several additional input fields, including: shaft centerline depth, inflow variation, ideal angle of attack, and the number of points over the chord. Additionally, two types of meanlines are available within the program: the NACA a=0.8 and the parabolic meanline. The thickness forms available include: NACA 65 A010, elliptical, and parabolic. OpenPVL is easily modified to accommodate additional meanlines and thickness forms.

18

Figure 2-5: *Single Propeller Design* **GUI.**

The GUI shows the following inputs:

- Number of Blades: 3
- Propeller Speed (RPM): 1600
- Propeller Diameter (m): 0.25
- Required Thrust (N): 625
- Ship Velocity (m/s): 5
- Hub Diameter/Diameter Ratio: 0.2
- Num. Vortex Panels over Radius: 25
- Max. Iterations in Wake Alignment: 10
- Hub Vortex Radius/Hub Radius: 1
- Number of Input Radii: 11
- Hub Unloading Factor: 0=Optimum: 0
- Tip Unloading Factor: 1=Reduced Loading: 0
- Swirl Cancellation Factor: 1=No Cancellation: 1
- Water Density (kg/m^3): 1000
- Shaft Centerline Depth (m): 3
- Inflow Variation (m/s): 0.3
- Ideal Angle of Attack (degrees): 1.54
- Number of Points over the Chord: 20
- Hub Image Flag (Check for YES): checked
- Meanline Type: NACA a=0.8
- Thickness Form: NACA 65A010
- Filename Prefix: OpenPVL
- Run OpenPVL_v1

r/R	c/D	Cd	Va/Vs	Vt/Vs	f0/c	t0/c	Skew	Xs/D
0.2	0.2269	0.008	1	0	0.0174	0.2056	0	0
0.25	0.2193	0.008	1	0	0.0195	0.1551	0	0
0.3	0.2148	0.008	1	0	0.0192	0.1181	0	0
0.4	0.2029	0.008	1	0	0.0175	0.0902	0	0
0.5	0.1885	0.008	1	0	0.0158	0.0694	0	0
0.6	0.1693	0.008	1	0	0.0143	0.0541	0	0
0.8	0.1545	0.008	1	0	0.0133	0.0419	0	0
0.9	0.1275	0.008	1	0	0.0125	0.0332	0	0
0.95	0.0801	0.008	1	0	0.0115	0.0324	0	0
1	0.0572	0.008	1	0	0	0	0	0

A single propeller can be designed and quickly evaluated graphically. One of the file outputs of the *Single Propeller Design* is the blade geometry. This feature of OpenPVL automatically transforms x, y and z coordinates of the designed propeller blade geometry into a command file that can be read by a CAD program. The user opens the command file in the CAD program RHINO™, and has the option of saving their design as a useful stereolithography (.STL) file, or as an Initial Graphics Exchange Specification (.IGES) file. Although the command file includes scripting specific to RHINO™, the propeller geometry can be exported to another CAD program after it is saved in RHINO™ as a file compatible to the other design software. The single propeller design option creates both propeller geometry and a corresponding scripting code which allows the propeller to be created as a .STL file and then printed on a 3D printer, as shown in Fig. 2-6. Propeller designs can be modified and saved with the OpenPVL tools repeatedly, at no cost to the user.
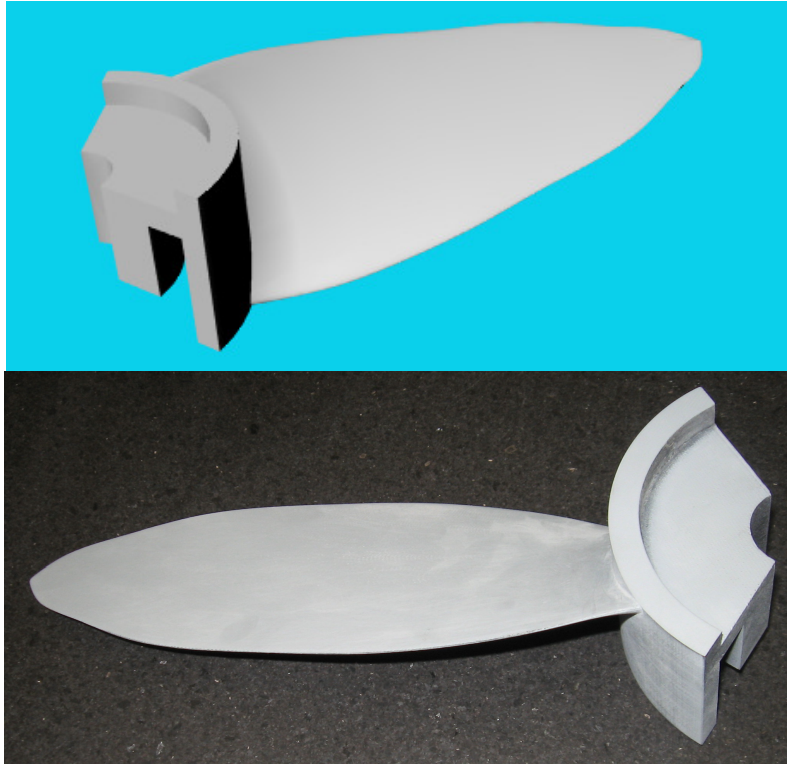
Figure 2-6: Propeller design in CAD (a) and the 3D printed result (b).

3 Propeller Printing: from Design Geometry to Prototype

3.1 Blade Geometry from OpenPVL to .STL Files

OpenPVL optimized the propeller design geometry, and then automated the creation of a RHINO™ command file. Using the RHINO™ NURBS modeling program, a blade with the desired geometry was drawn, converted into a .STL file, and was ready to send to a three-dimensional printer, as detailed in Chapter 5. OpenPVL uses several simple steps to guide propeller users to rapidly create optimal propeller geometry that can be printed, tested, and ultimately put to use.

OpenPVL allows users with limited backgrounds in propeller design to produce a prototype propeller. In this study, OpenPVL created a file readable by a 3D printer, as illustrated in Fig. 3-1. First, propeller characteristic boundaries were designed with the *Parametric Analysis* tool in OpenPVL. Desired characteristics of the vehicle were entered into the *Parametric Analysis* GUI of OpenPVL in order to determine the range of appropriate propeller characteristics that should be used as input for the *Single Propeller Design* of OpenPVL.

Next, desired design characteristics were entered in the *Single Propeller Design* option of OpenPVL. The user has the option to create a filename and save a variety of propeller initial inputs and outputs, including geometries that can be read as command files into the RHINO™ CAD program. RHINO™ has the capability to convert the geometry created by the command file into saved .STL and .IGES files, among others. The .STL files can be read and printed into model blades, from which molds may be made. The .IGES files are recognized by a variety of CAD programs.
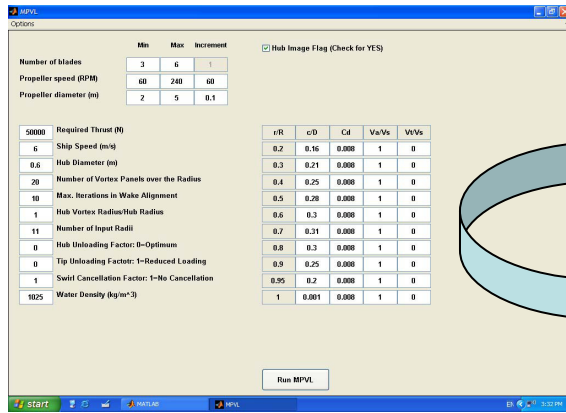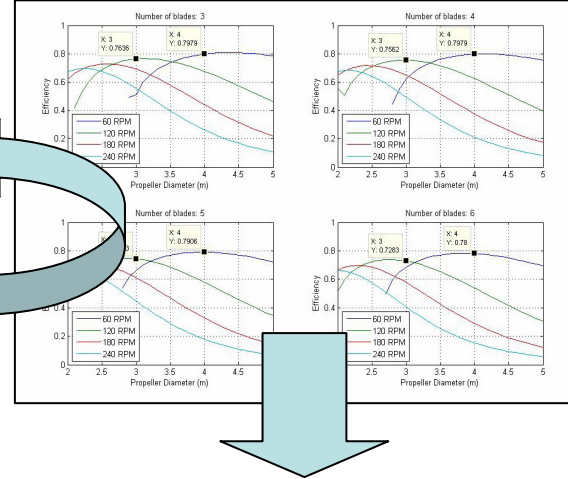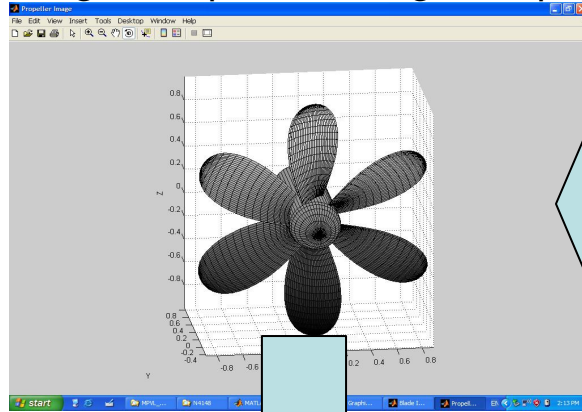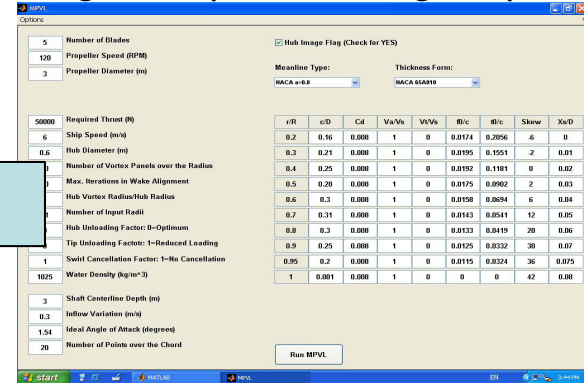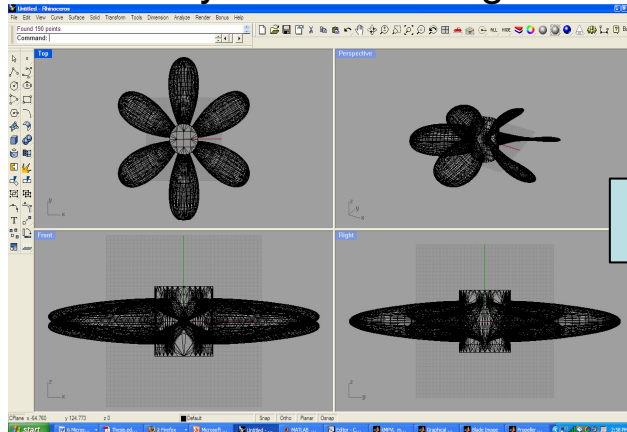
Figure 3-1: OpenPVL design-to-prototype process.

3.2 Fusion Deposition Modeling (FDM) Printer Overview

There are several different types of machines and manufacturers of 3D printing machines, such as FDMs (StrataSys™™ and Solidscape™), stereolithography machines (3D Systems™), and printers capable of making 3D design parts or can print a mold (Z-Corp™). All of these machines accept the .STL file format as input. The method of 3D printing available for this study was with an FDM.[2] FDM printing requires the input of a .STL (Stereolithography) file. Specifications of the 3D printer used in this study are listed in Table 3-1 [11].

Table 3-1: StrataSys™ Titan FDM Specifications.

Build Volume	Within 16 inches by 14 by 16 inches (LxWxH).
Materials	acrylonitrile butadiene styrene (ABS), polycarbonate (PC), and PC-ABS
Layer thickness	Ranges from 0.007in to 0.013in
Support structures	ABS and PC-ABS are water-soluble
Material Canisters	Two 92in for the build materials and the support materials
Labor	Machine may be unattended during builds, but requires extra labor when changing material type used in the machine
Software	.STL files are imported using Insight™

Once a .STL is input into the FDM printer software, a support structure is automatically created for the input file, as shown in Fig. 3-2. The support structures of ABS and ABS/PC both are built such that they dissolve after printing, during the finishing process.   The support structure of polycarbonate must be filed down manually, which leaves room for deviation from the intended geometry, as well as more time invested in the manufacturing process.

---

[2] The specific FDM used in this study was a StrataSys™ for the Titan FDM.
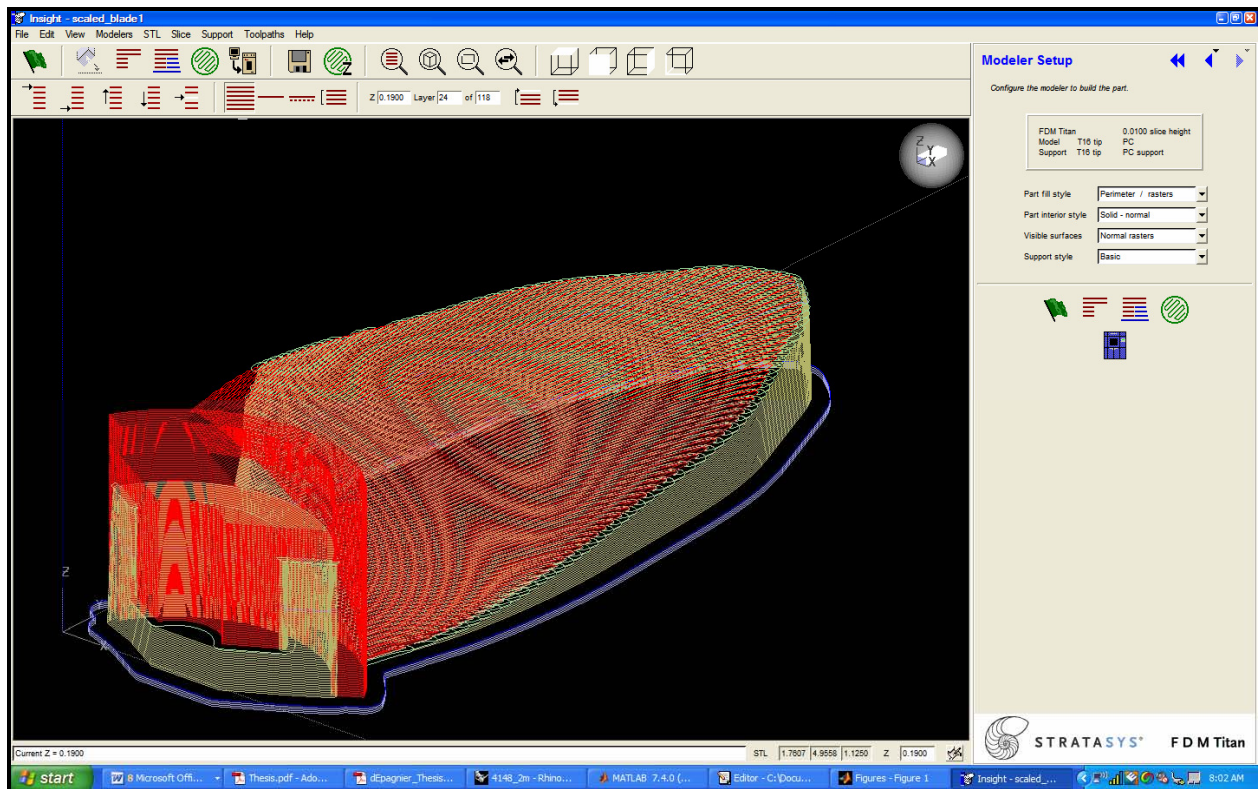
Figure 3-2: Screenshot of the propeller blade and support structure generated by the StrataSys™ Insight software. This is the design blade described in more detail in Chapter 5.

The FDM printable file generates both a total volume calculation, as well as a printing completion time estimate. The total volume calculation is used by some FDM printer operators to determine the price per volume of the object printed. The time to print the volume of both the propeller and its support structure was modeled for various diameters, as in Fig. 3-3. The plot indicates that the time needed to print a 25 cm three-bladed propeller prototype takes approximately 44 hours.
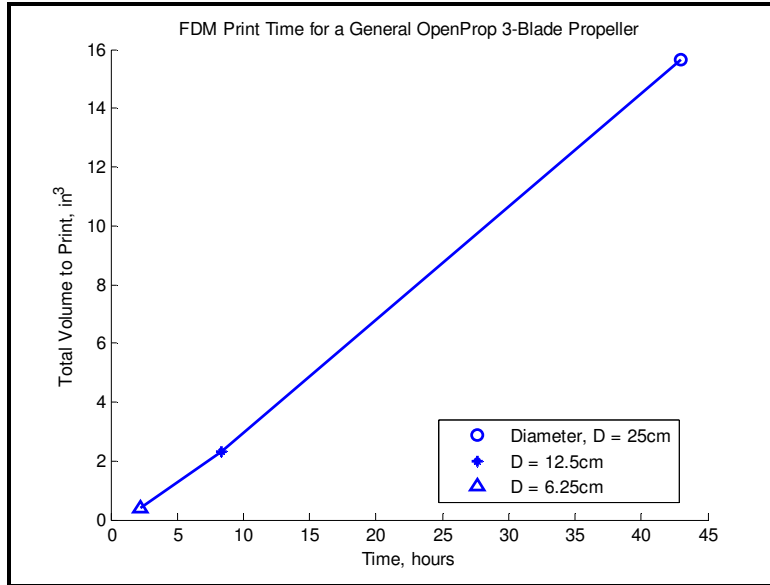
Figure 3-3: Time to print estimate for propellers of varying diameters.

3.3 Analysis of Materials for Fusion Deposition Modeling

Material options for a Fusion Deposition Modeling (FDM) printer include three types of plastics: acrylonitrile butadiene styrene (ABS), polycarbonate (PC), and an ABS/PC blend. While ABS solids can be easily filed down to a smooth finish, if necessary, after printing, the material's tensile and flexural strengths and modulus of elasticity are not comparable to that of the ABS/PC material. Additionally, the polycarbonate and the ABS/PC are more rigid than ABS, and therefore undergo less deformation from the desired propeller prototype geometry. A summary of material options for this particular Titan FDM™ is in Table 3-2.

Table 3-2: Summary of Propeller Materials [11,12].

Material	% Water Absorption (24hrs)	Hardness Rockwell & Burnell, 73°F	Specific Gravity	Tensile Modulus of Elasticity, 73°F (PSI)	Flexural Modulus of Elasticity, 73°F (PSI)
ABS	0.3	R105	1.05	236,000	266,000
PC	0.2	R115	1.20	290,000	310,000
ABS-PC	—	R110	1.20	265,000	270,000

The printed propeller blade geometry can also be used to cast a mold. Using this method, propellers can be manufactured from hard polyurethane, or other substance which affords greater loads to be subjected upon the propeller blades both in prototype testing and for use on underwater vehicles. The additional benefit to constructing a mold is that it is more economical than using an FDM machine, and multiple molds can be made, decreasing the manufacturing time.

The three propeller blades with their connecting attachments took an estimated 91 hrs 4min to print. The printing process at 29 hours is shown in Fig. 3-4. In order to reduce both time and cost spent, the blades were printed with the female side of the connection to the hub facing upwards, which decreased the amount of FDM support structure.
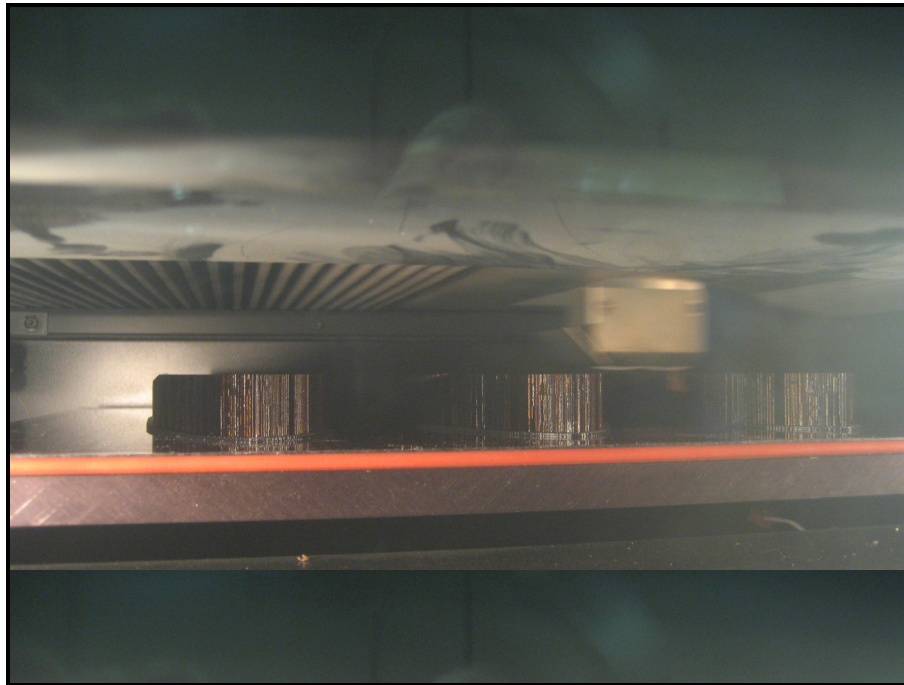
Figure 3-4: FDM printing at 29 hours.

Once the propeller blades are printed, the support structure is dissolved in a chemical bath of soluble concentrate[3]. The bath containment vessel also produces vibrations, which aid in the removal of the support structure. After the propeller parts are removed from the bath and

---

[3] The chemical bath used was WaterWorks soluble concentrate P400SC, in a one part concentrate to eleven parts water ratio.

dried, the blades require smoothing. A combination of sandpaper and filler primer can achieve a smooth finish[4]. The primer fills in the gaps among the composite material threads used by the FDM printer for blade production.

---

[4] Dupli-Color high-build formula primer can be used.

4 Studies on an Off-The-Shelf Propeller for the SeaBED class AUV[5]

The effectiveness of off-the-shelf propellers was determined experimentally in this case study. The propellers were tested to match their desired operational vehicle speed while maintaining speeds appropriate to their thruster pairings. The Bollard test was used for the purpose of testing how great a thrust the propeller could produce for a given propeller blade RPM while at rest. The mobile cart test was used to simulate the propeller thrust capacity at operational speeds.

The vehicle modeled in this case study was a low-speed, high-drag underwater vehicle. Desired thruster rotational speed ranged from 60 – 120 Revolutions Per Minute (RPM). Cosmos FloWorks™, a fluid-dynamic software, calculated the vehicle's necessary thrust for a single propeller at a design speed of 1m/s. From the design point of total thrust required at 1 m/s, the thrust needed for other points can be extrapolated from the relationship $Drag \propto Velocity^{3}$, as in Fig. 4-1 [13].

---

[5] The SeaBED AUV is one of several in its class developed in the Deep Submergence Lab at the Woods Hole Oceanographic Institute.
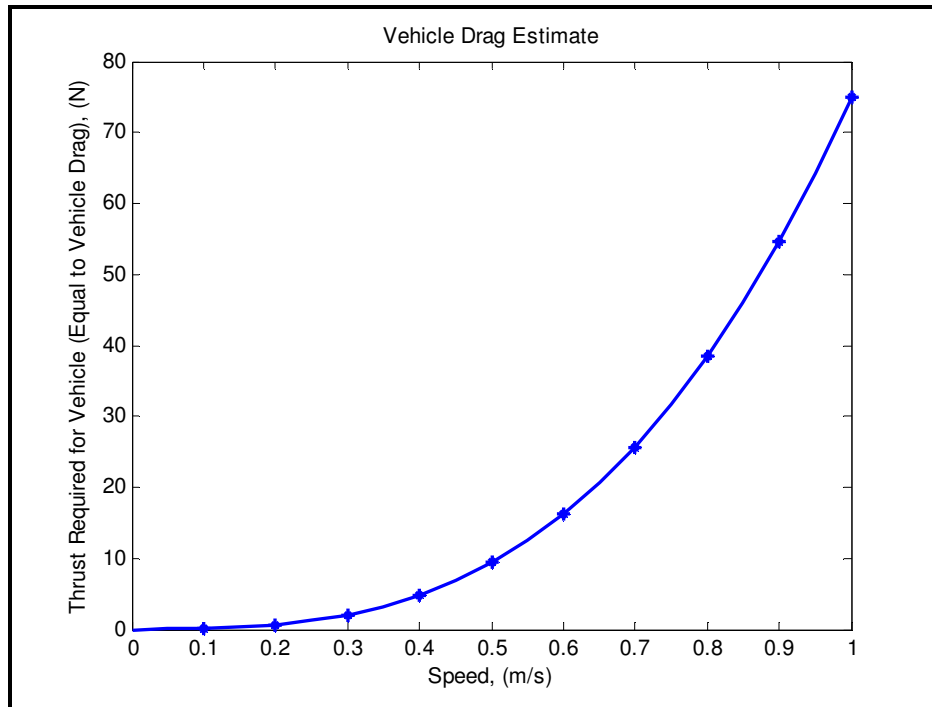
Figure 4-1: Required thrust estimate for a low-speed, high-drag AUV.

4.1 Experimental Set-up

The 20m tow-tank at Woods Hole Oceanographic Institute was used for experimentation. The majority of the set-up was on the tow-tank carriage platform (Fig. 4-2).
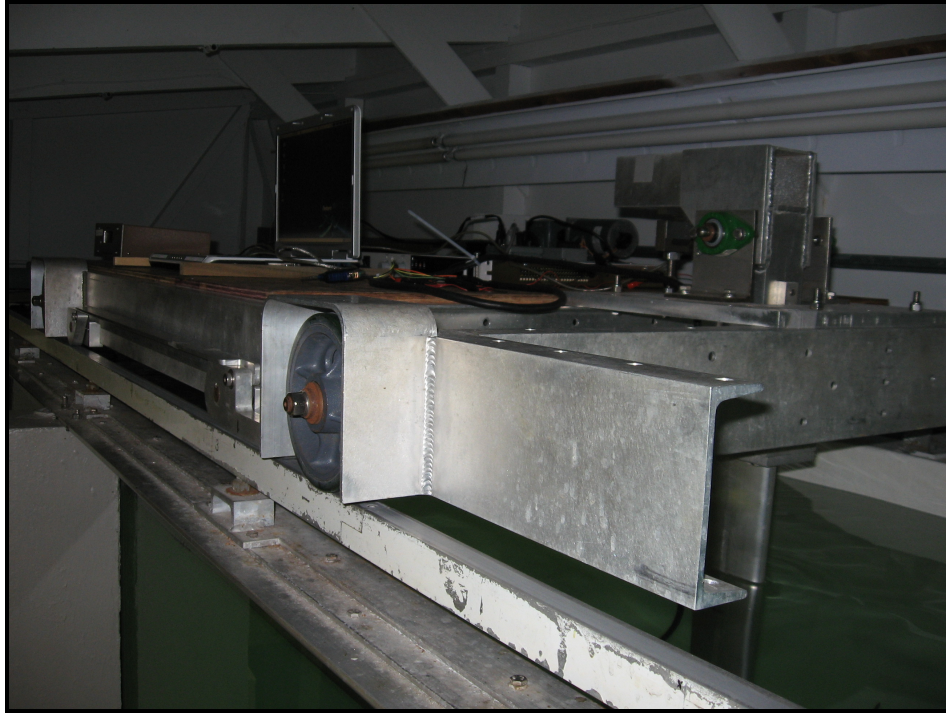
Figure 4-2: Tow-tank carriage.

The following instruments were secured to the top of the carriage:

- Laptop computer
- Power supply
- Load cell
- Thruster mount with thruster and propeller attached

The computer used a Linux program for recording the voltage and current draw of the thruster motor at a specified RPM. The power supply was set at 48V, and was monitored for possible spikes in current when first powered. The load cell was attached in compression to the top of the thruster mount, and measured the thrust force of the propeller. The thruster mount used a foil shape to minimize external effects of turbulent flow on the testing of the propeller. The propeller and thruster attached underneath the thruster mount using three hose clamps (Fig. 4-3).
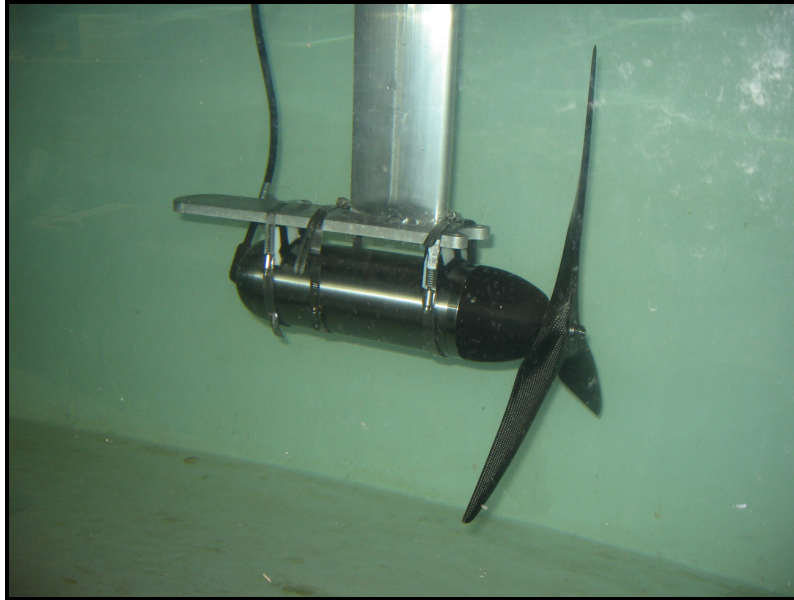
Figure 4-3: Thruster mount.

The computer used a program which controlled propeller RPM while recording the thruster current and voltage over the time of the test run.

4.2 Method for Bollard and Mobile Cart Test Analysis

A set range of data values was averaged, so that the data analyzed was taken at the same time for all runs. This simplifies the process, and eliminates data that could be skewed towards the end of the run. A brief analysis showed the current and voltage remained fairly constant throughout a run, making the process of truncating and averaging data a good measurement of current and voltage.

The force measurement recorded was taken during the middle of the run, at the point at which the cart had gained full momentum, but before it had started to slow down. The cart was programmed such that the cart would accelerate and decelerate to and from the desired speed linearly, for a desired distance. Both desired cart speed and total distance of the run were input into the cart's computer. Propeller rotation values ranged from 16.65RPM to 133.2RPM, and for these values, current and voltage were recorded by machine.

31

4.3 Bollard Test Results

The Bollard test results are shown in Fig. 4-4 and 4-5. The tow-tank testing provides a comparison among different variables that are independent of RPM, including: current, A, (Amps), voltage, V (Volts) and thrust (N), as in Fig. 4-4. Fig. 4-5 compares the propeller thrust (N), the propeller rotational speed (RPM) and the power input to the thruster motors (J/s).

As expected, the propeller voltage remained close to the set voltage, at ~48 Volts. However, the current draw acted as an indicator of the upper end of the propeller's performance. The current draw was at (O)0.1 Amps for the lower propeller speeds tested, and increased exponentially for the highest speeds, to (O)1. As propeller rotational speed increased, thrust values increased exponentially, with a maximum value of thrust recorded as 64.5 N at a propeller speed of 133 RPM. Numerical values for the Bollard Test and the Moving Cart Test can be found in Appendix B.4. Fig. 4-5 provides an estimate of the power consumption required by the propellers in order to achieve propeller speeds from 0-133 RPM.
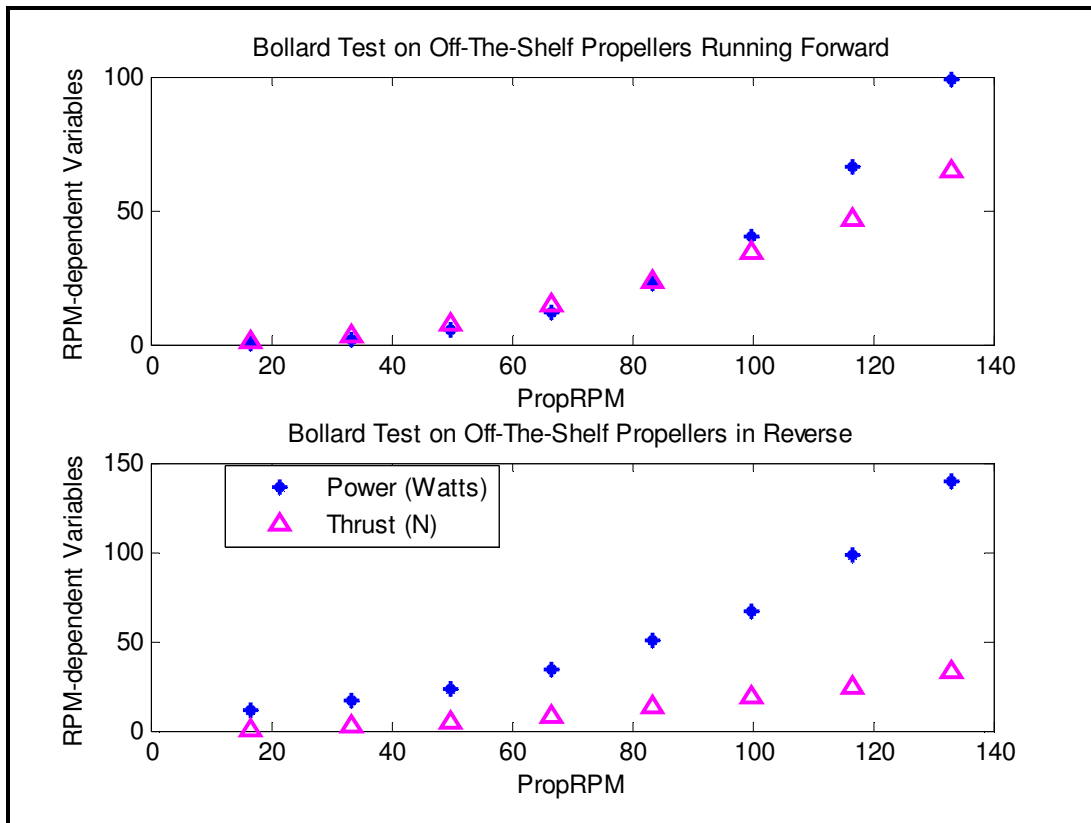
Figure 4-4: **Off-the-shelf propeller Bollard Test results.**
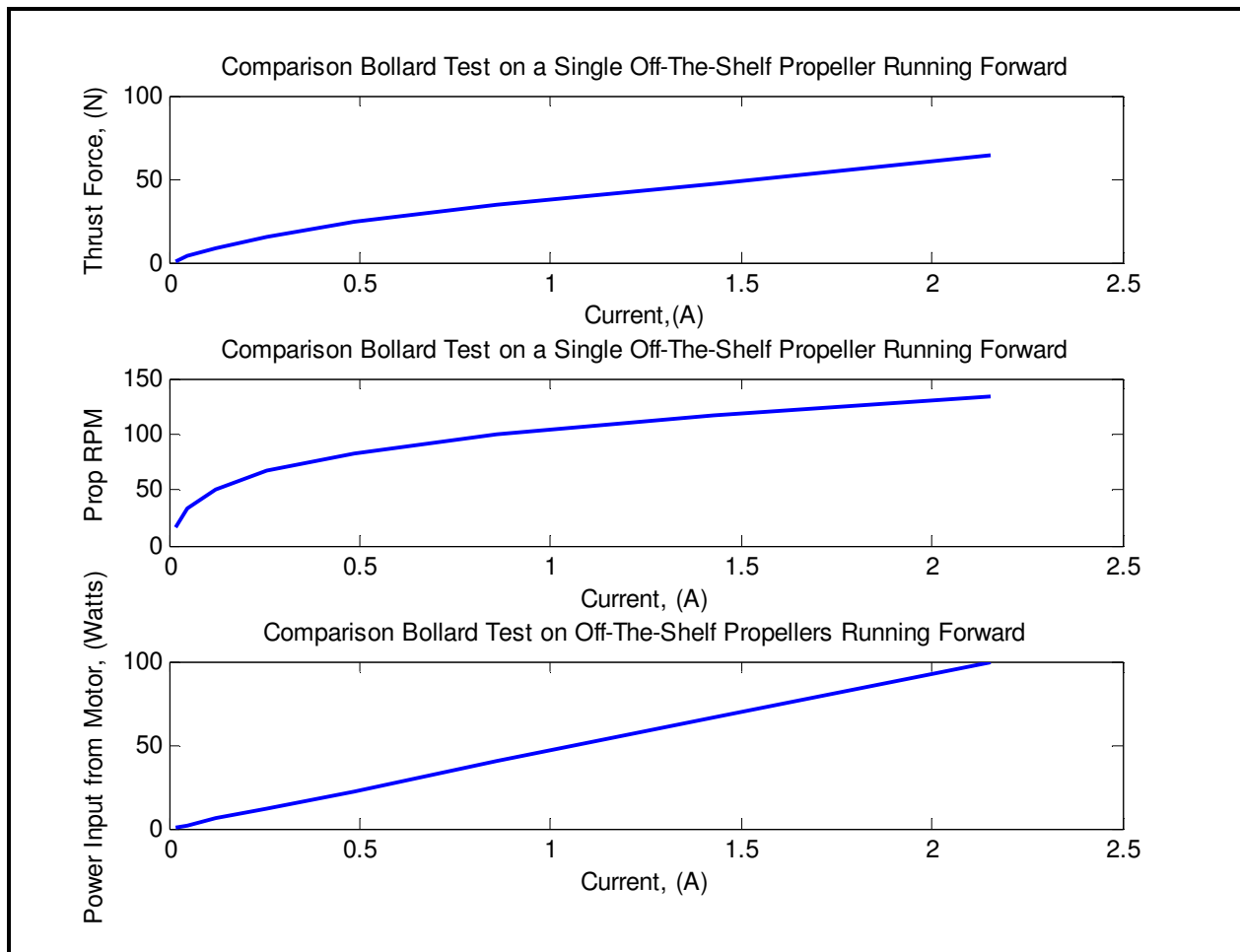
Figure 4-5: Bollard Test comparison of thrust, propeller RPM and power input with respect to current.

4.4 Mobile Cart Results

The mobile cart tests compare the propeller thruster current and voltage intake and the thrust force produced, as summarized in Fig. 4-6. Each diagonal line of data for the thrust corresponds to a set propeller RPM. The voltage remained at a constant 48 V throughout the test runs, while the current increased exponentially from (O)0.1 to (O)1. The maximum thrust occurred at a cart-simulated speed of 0.25 m/s, with a value of 61.88 N for a propeller rotational speed of 133 RPM. The current, and therefore power requirements, increase significantly for rotational values greater than the 99.9 RPM test series (Appendix B.4). Fig. 4-6 super-imposes the thrust requirements shown in Fig. 4-1, to clearly present the tested propeller's range of operation. Fig. 4-7 summarizes the off-the-shelf propeller performance using K_T vs. J curves.
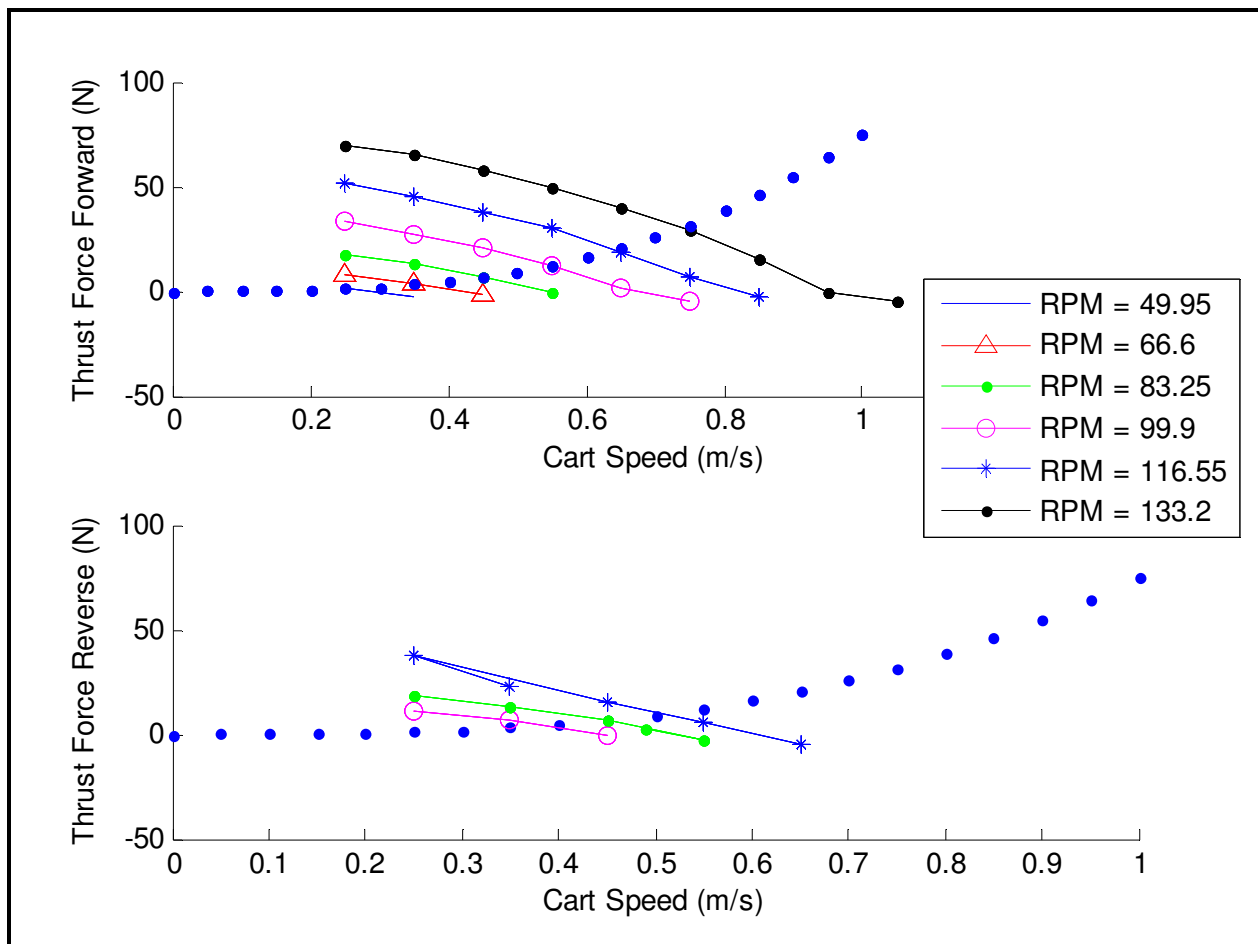
Figure 4-6: Moving Cart Test results. All data ABOVE the dotted line meet the calculated drag requirements for the vehicle.
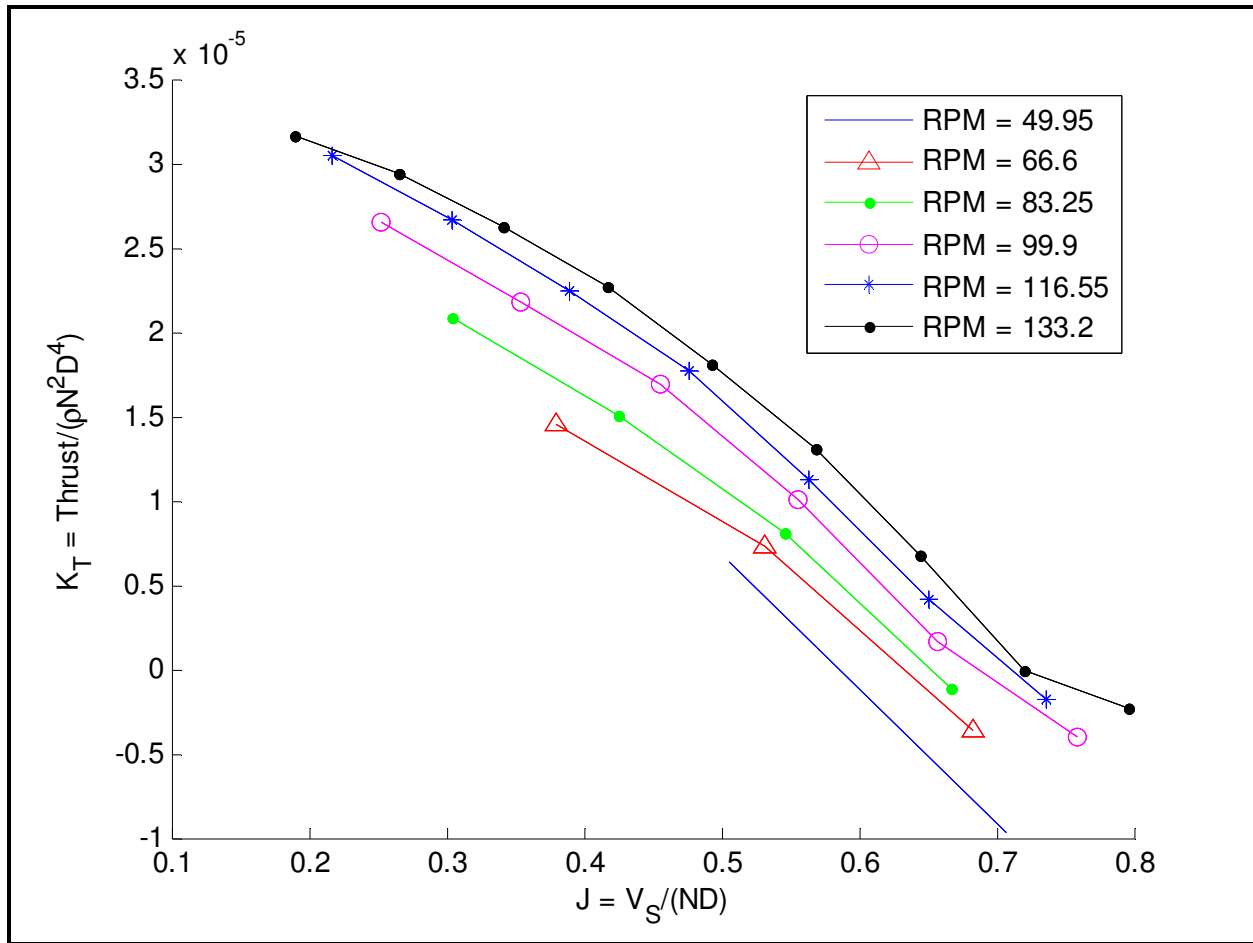
Figure 4-7: Off-the-shelf propeller forward Mobile Cart test results, thrust coefficient, K_T vs. the advance coefficient, J.

The value of maximum speed for the forward-moving propeller must not compromise the propeller thruster limitations or exceed reasonable power requirements. Therefore, the highest speed point of interest is that which satisfies propeller speed, power and thrust requirements at 0.55m/s for forward propulsion and at 0.35 m/s for reverse. Overall, the thruster tests indicate that the propeller does not meet requirements for a design speed of 1m/s. Therefore, a propeller will be designed and produced to meet the vehicle operational requirements. A detailed description of this AUV propeller design is presented as a case study in Chapter 5.

5 Case Study: Designing and Producing a SeaBED Class AUV Propeller

Vehicles carrying large quantities of instrumentation, moving at relatively slow speeds fall under the low-speed, high-drag category of AUVs. The main purpose of the study is to create a propeller blade for optimal performance for the design characteristics of a test case AUV. Another object of this research is to determine the effects of C_L on an off-the-shelf propeller design.[6] OpenPVL and basicPVL provide the means to analyze the required design characteristics of such an AUV, as shown in the remainder of this chapter. In this study the design codes were used to:

- Estimate the performance of the off-the-shelf propeller.
- Design a custom propeller for the vehicle.

User input parameters can be derived from an off-the-shelf propeller design, and the code used to estimate the performance of these propellers. Thus, the propeller design tool can be used to guide the designer to an appropriate off the shelf design.

5.1 Test Case AUV Characteristics

The off-the-shelf propeller used in the test-case AUV was modeled for efficiency, and it had the following characteristics:

- Tests showed that the propeller operated on this vehicle at 170RPM in order to reach a speed of 1.0m/s, and at 60RPM for a speed of 0.3m/s. Since the desired range of RPS was roughly between 1-3, 2 RPS was be taken as the target rotational speed.
- The propeller had three blades, or z = 3.
- The hub was 0.031m, and was 0.0104 in terms of r/R.

---

[6] The 'off-the-shelf' propeller used by the AUV in this study is a 24in. carbon fiber model airplane propeller developed by Aircraft International Carbon Propellers.

- The diameter, D, was 0.594.
- The chordlength distributions, c/D with respect to r/R, where r was the location on the radius and R was the radius length, were measured using calipers, as in Table B-1.
- Due to the placement and size of the thrusters, the axial flow ratio was equal to one for this particular AUV.

To validate the propeller design process, a test vehicle was used. Test vehicle characteristics implemented in the propeller design included [13,14]:

- Design speed, Vs, was 1m/s.
- Inflow wake velocity variation, Vvar =/-0.03m/s.
- Design depth was 200m to 1km.
- Cavitation effects were tested for a depth of d = 2m.
- Propeller type was assumed free tip double screw.
- The required forward thrust at operational cruising speed for the AUV was 150N, or 75N per propeller.
- Tests taken from thrusters of a previous AUV show that it was most acceptable to model the propeller at 2RPS.
- Thruster diameter was small relative to the propeller diameter, at 4.25in. or 0.1080m, and was ignored.
- Hull length, L = 96in., or 2.438m.

5.2 Blade Selection

The number of blades used on the propeller was chosen by varying parameters of the blade number, diameter, and revolutions per second in m-file basicPVL.m. The blade numbers analyzed ranged from 3 to 6. The diameter was varied from 0.5 to 0.9m, since the values of the advance coefficient, J, were within an acceptable range of 0.3 to 1.9. Rotational speeds appropriate for the analyzed thruster range from 2 to 3 RPS. The results were assessed using enumerated contour plots, which were produced for each blade number, and varied contours of the efficiency η, with the ranges of diameter, D (in the y-axis), and revolutions per second (RPS), N (in the x-axis). Figure 5-1 shows a contour plot for the chosen number of blades, 3. Plots for 4, 5 and 6 blades can be found in Appendix B.3.
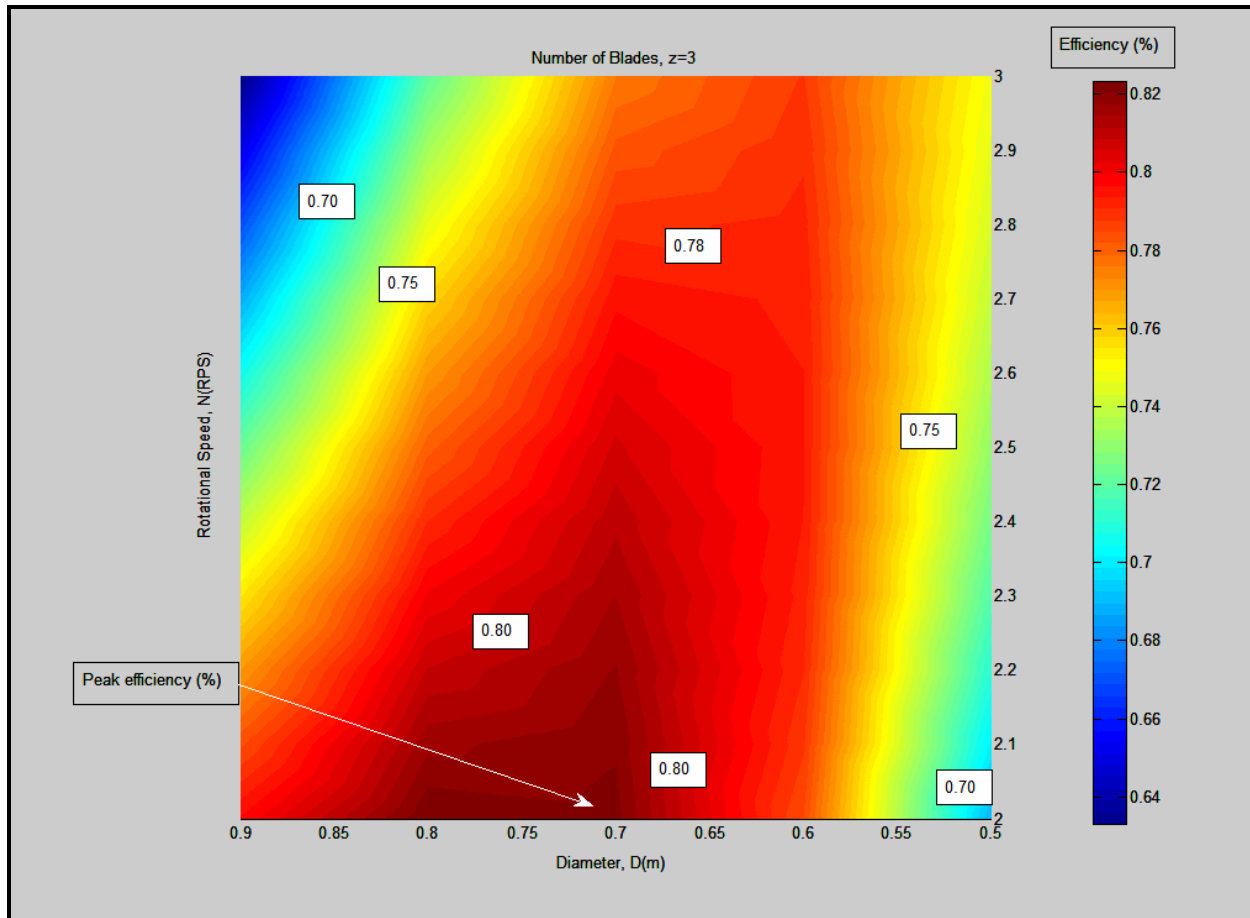
Figure 5-1: Efficiency for a 3-bladed propeller.

The initial design exhibited three blades, since a three-bladed propeller is more efficient in *all ranges* of diameter and rotational speed than the others. This is because a larger number of blades with a smaller diameter provide greater efficiency for the lowest range of N, whereas the three-bladed propeller provides the most efficiency at 2 RPS, which coincides with the design speed 1.0m/s [14]. Furthermore, the three-bladed propeller had a wide range of high efficiency (70% or more) at all ranges of desirable values for N. It is interesting to note that the maximum efficiency on the 3-blade propeller was for smaller values of N. The blade chosen that best satisfies all the desirable values of N, is 0.625m in diameter, with an initial calculated efficiency of 0.7091.

5.3 Propeller Performance Characteristics Comparison

A comparison between both the initial and final propeller design was made. This helped determine how to improve the existing propeller, as well as to provide guidance on purchasing future off-the-shelf propellers. Detailed blade performance characteristics and blade geometry are shown in figures 5-1 and 5-2 respectively. Figure 5-3 and 5-4 show the performance and geometry for the equivalent off-the-shelf type propeller. Note that the off-the-shelf equivalent propeller required section lift coefficients of about 1.0 which is close to stall and would have high drag. Therefore, the analysis indicates that the off-the-shelf propeller is overloaded for this application. It is important to note that the table values used optimal calculations for the pitch over diameter, and these values were significantly different from the actual P/D, which is measured as 0.427 for r/R equal to 0.7. The data for the final propeller design, in which lift and non-cavitation requirements were easily fulfilled, is shown in Table 5-1.

Table 5-1. Final Optimized Propeller Performance vs. Radius Position

r/R	V*	B	Bi	C_L	G	Σ	$-C_{Pmin}$
0.2000	1.2788	51.8538	57.9439	0.4936	0.0228	43.7157	0.56
0.2500	1.4060	45.5278	51.9462	0.4871	0.0239	118.7362	0.55
0.3000	1.5478	40.3256	46.7912	0.4867	0.0258	97.8371	0.53
0.4000	1.8621	32.4814	38.6047	0.4866	0.0293	67.4235	0.49
0.5000	2.2023	26.9896	32.5685	0.4754	0.0314	48.0748	0.48
0.6000	2.5581	22.9968	28.0254	0.4599	0.0317	35.5383	0.46
0.7000	2.9242	19.9909	24.2683	0.4075	0.0293	27.1253	0.38
0.8000	3.2973	17.6570	20.8232	0.3519	0.0235	21.2771	0.33
0.9000	3.6749	15.7984	17.5931	0.3120	0.0146	17.0847	0.27
0.9500	3.8645	15.0050	16.0380	0.2578	0.0091	15.4287	0.24
1.0000	4.0543	14.2864	14.5165	0.2391	0.0005	13.9991	0.22

In this design, the rake/D and the skew were assumed to be zero. The ratio of pitch to diameter, P/D, was calculated using Equation 5.1 [4]:

$$P\!\!\Big/\!\!_D = \tan(B_i + \alpha_i)\pi\left(\frac{r}{R}\right),$$ where α_i = 1.54°, from NACA Mean Line a = 0.8.   **5.1**

The design requirements are summarized into Tables 5-2, 5-3 and 5-4, from which the optimized propeller was fabricated.

Table 5-2: Blade geometry for optimized propeller, using ML type NACA, a=0.8, and NACA-66 sections.

r/R	P/D	c/D	Δα	fo/c	to/c	rake/D	Skew
0.2000	1.0660	0.2269	2.6878	0.0335	0.079	0	0
0.2500	1.0609	0.2193	2.4448	0.0331	0.078	0	0
0.3000	1.0590	0.2148	2.2207	0.0330	0.070	0	0
0.4000	1.0599	0.2029	1.8460	0.0330	0.055	0	0
0.5000	1.0638	0.1885	1.5609	0.0323	0.050	0	0
0.6000	1.0693	0.1693	1.3438	0.0312	0.046	0	0
0.7000	1.0635	0.1545	1.1756	0.0277	0.041	0	0
0.8000	1.0340	0.1275	1.0426	0.0239	0.039	0	0
0.9000	0.9809	0.0801	0.9355	0.0212	0.030	0	0
0.9500	0.9455	0.0572	0.8896	0.0175	0.025	0	0
1.0000	0.9042	0.0032	0.8479	0.0162	0.023	0	0

Table 5-3: Off-the-shelf propeller performance vs. radius position.

r/R	V*	B	Bi	C_L	G	Σ	$-C_{Pmin}$
0.2000	1.0888	68.7862	73.2997	1.0691	0.0143	198.7506	0.93
0.2500	1.1767	59.1563	65.2233	1.2170	0.0169	169.9533	1.02
0.3000	1.2923	51.1250	58.0726	1.3675	0.0211	140.6989	1.12
0.4000	1.5825	39.2448	46.5828	1.3573	0.0289	93.5663	1.09
0.5000	1.9215	31.3387	38.2319	1.3741	0.0342	63.2903	1.05
0.6000	2.2877	25.8917	32.1280	1.3561	0.0365	44.5251	1.02
0.7000	2.6704	21.9756	27.2771	1.2697	0.0349	32.5871	0.95
0.8000	3.0637	19.0490	23.0115	1.1103	0.0289	24.6877	0.81
0.9000	3.4634	16.7904	19.1435	0.8699	0.0185	19.2633	0.63
0.9500	3.6646	15.8456	17.3195	0.6695	0.0117	17.1820	0.49
1.000	3.8661	14.9993	15.5565	0.6255	0.0007	15.4153	0.46

Table 5-4: Blade geometry for the off-the-shelf propeller, using ML type NACA, a=0.8, and NACA-66 sections.

r/R	P/D	c/D	Δα	fo/c	to/c	rake/D	Skew
0.2000	1.2058	0.0770	3.1565	0.0726	0.067	0	0
0.2500	1.1707	0.0742	2.9210	0.0826	0.061	0	0
0.3000	1.1572	0.0751	2.6597	0.0929	0.053	0	0
0.4000	1.1494	0.0846	2.1720	0.0922	0.047	0	0
0.5000	1.1505	0.0814	1.7890	0.0933	0.037	0	0
0.6000	1.1551	0.0739	1.5026	0.0921	0.032	0	0
0.7000	1.1476	0.0647	1.2873	0.0862	0.028	0	0
0.8000	1.1137	0.0533	1.1221	0.0754	0.023	0	0
0.9000	1.0532	0.0386	0.9926	0.0591	0.021	0	0
0.9500	1.0130	0.0301	0.9381	0.0455	0.019	0	0
1.0000	0.9663	0.0017	0.8892	0.0425	0.018	0	0

It is very important to note that the actual pitch over diameter ratio for the existing propeller was about a third of the optimal pitch calculated for blades of its geometry. The 2D analysis determined that the lift coefficients on the existing propeller were above 1.0 for about 85 percent of the length the of propeller's radius.

5.4 Off-the-Shelf Propeller Analysis and Improvement

Lift coefficients above 1.0 may result in stalling effects, and therefore are unacceptably high. Therefore, appropriate analysis included determining the effect of hub diameter variation on the lift coefficients, while still considering the effect of hub variation on propeller efficiency, as shown in Fig. 5-2. Several assumptions in this study are of note: the efficiency does not consider the possibility for lost performance due to stalling from high lift coefficients. Additionally, the performance modeling assumes a NACA camber, thickness, and a calculated pitch to diameter (P/D) ratio.
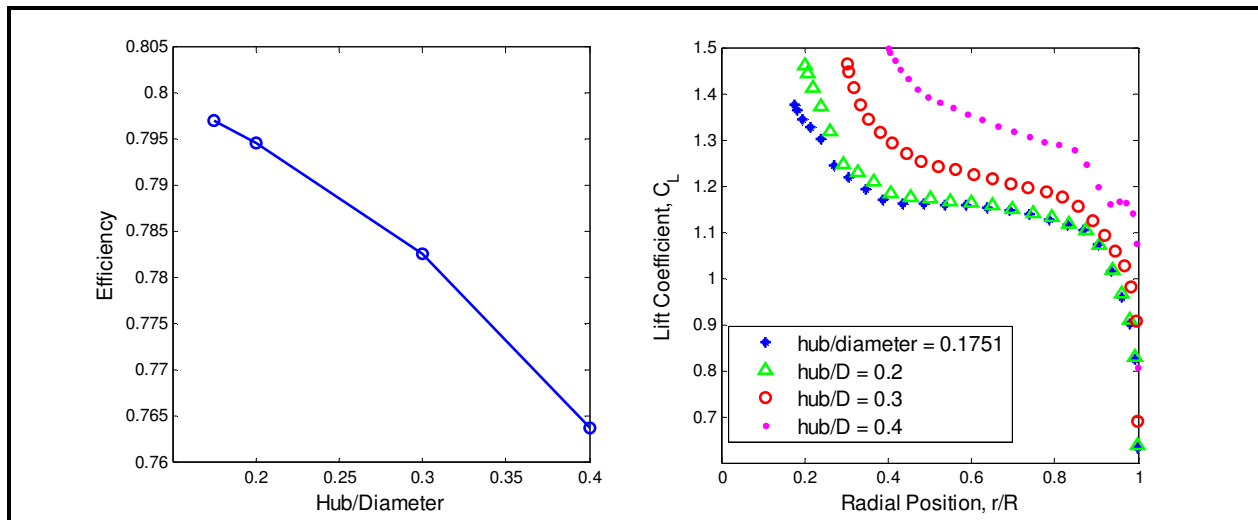
Figure 5-2: Efficiency and C_L for varying hub diameters and z = 3blades, N = 2RPS, D = 0.594m.

One experimental problem for this study was to determine whether adding a fairing to increase hub size would be performance enhancing. From Fig. 5-2, it is clear that the efficiency decreased with an increase in the hub radius. As the hub size increased, so did the lift coefficient; attaching a fairing to reduce the effects of stalling was not applicable in this case. Therefore, the propeller geometry must increase in diameter or chordlength to reduce the lift coefficients enough to avoid stalling.

41

Due to high values of lift coefficients for the model-airplane propeller, the effect of different blade diameters on lift coefficient and calculated efficiency were modeled. A list of criteria determined the chosen propeller design:

- The lift coefficient would be reduced to 0.7 or below at the hub, and below 0.25 at the blade tips.
- The efficiency would be maximized for $z = 3$ and $N = 2$, with the original chord distribution.
- The optimal diameter would be as close as possible to the original diameter, while satisfying the design requirements.

The chordlength distribution which satisfied these requirements can be found using basicPVL. The numerical values for the chordlength distribution is in Appendix B.1.
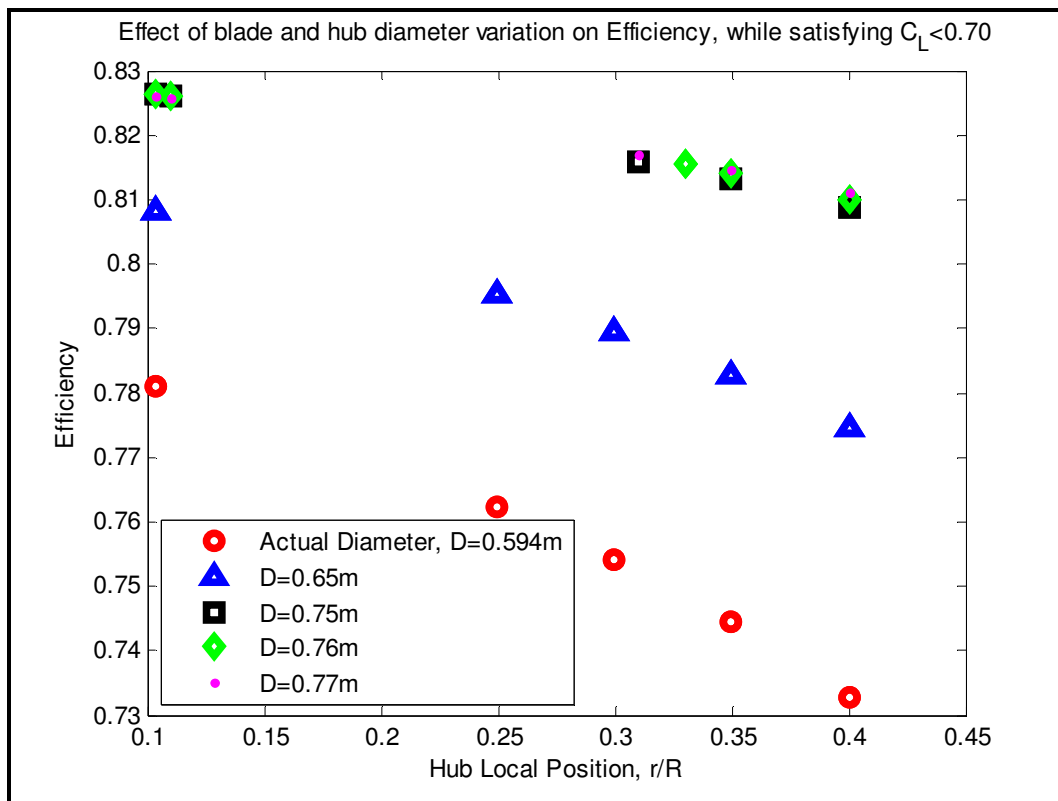
Figure 5-3: Efficiency vs. variation of blade hubs for different diameters of the off-the-shelf propeller. In this case, $C_L<0.7$ is satisfied for D = 0.76m and 0.77m.

Using the results shown in Fig. 5-3, the design chosen was 0.76m, with a hub extending to r/R = 0.31, and an efficiency of 0.8154 and an actuator disk efficiency [4] of 0.9287.   From Figs. 5-2 and 5-3, it is clear that an increased hub (proportionally greater than r/R = 0.2) has a greater benefit for larger propeller blades (greater than 0.6m) which maintain lower rotational speeds (less than 2 RPS).   Due to the relatively low RPSs needed for this vehicle and the operating depth, cavitation is not likely to occur [6].   However, the case in which cavitation could present problems will be investigated by analyzing propeller pressure coefficients at a shallow operating depth of 2m.   A visual comparison of the off-the-shelf propeller modification and the final propeller design is presented in Figs. 5-4 and 5-5.
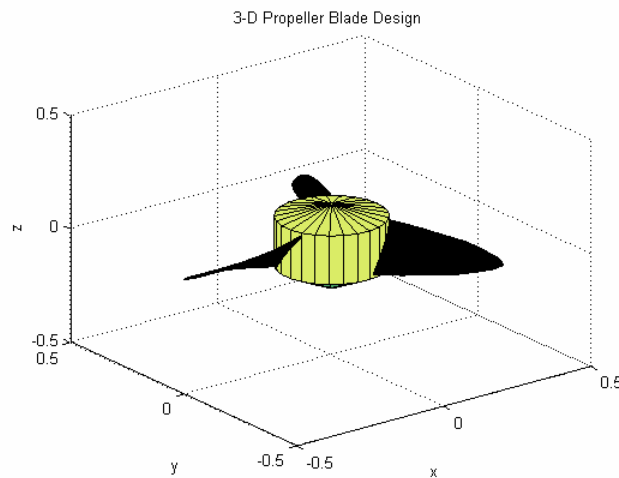
Figure 5-4: MATLAB™ representation of optimal modified off-the shelf propeller.

5.5 Propeller Hub Assembly and Shaft Design

The hub was machined out of Delrin. The new propeller hub, (Fig. 5-6), allowed for the three individually-printed propeller blades to tightly slide into place. The unique propeller blade design allowed for the 3D printing of a propeller with a diameter that exceeded the dimensions of the FDM machine.
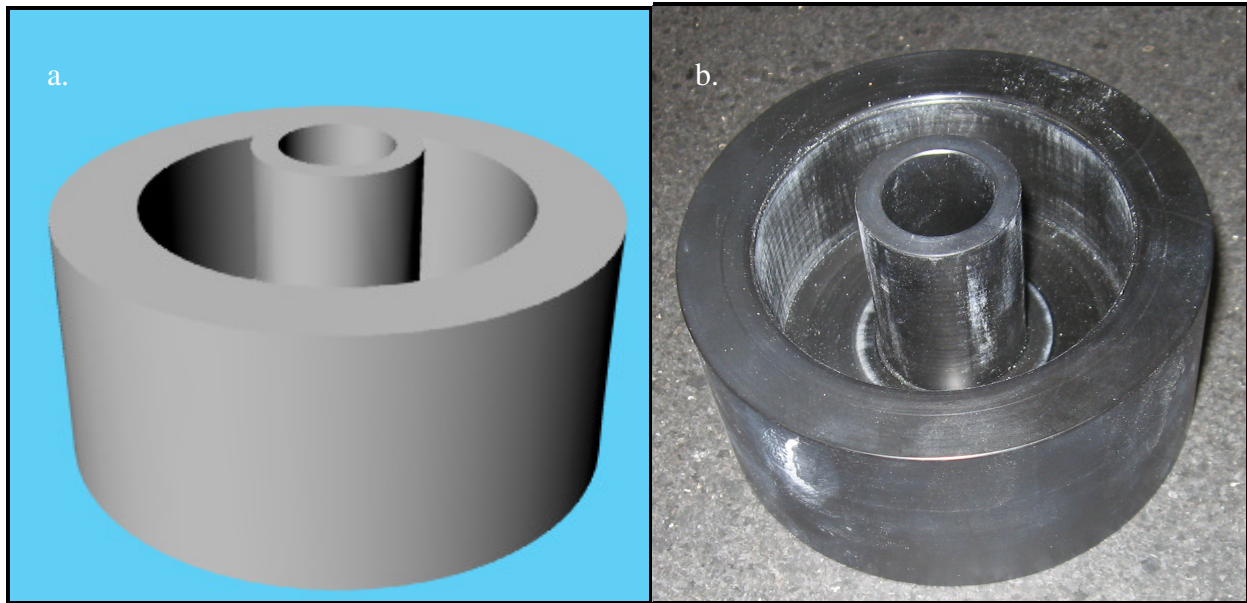
Figure 5-6: Hub design in CAD (a), and machined (b).

The propeller design is useful only if it is integrated into the design of an AUV. One of the necessary modifications is to adjust the length of the propeller shafts to accommodate the new propeller blade pitch. The purpose of the thruster shaft is to transfer the energy of the thruster motors to the propeller blades. The optimized propeller design used a pitch to diameter ratio of 1.0635 (for r/R = 0.7), while the original design allowed for a propeller with blades with a lower pitch to diameter ratio of 0.427 (for r/R = 0.7).

The original propeller thruster shaft was modified to work with a newer, more highly-pitched propeller. The new propeller shaft design can be seen in Fig. 5-7. The shaft was machined out of Grade 5 Titanium. Dual-usability of the new shaft allowed for the new propeller to be attached using a pin and bolt with a lock-washer mechanism. For ease of test comparison

to the off-the-shelf propellers, the shaft contained a pinhole to accommodate the width of the original propeller hub.
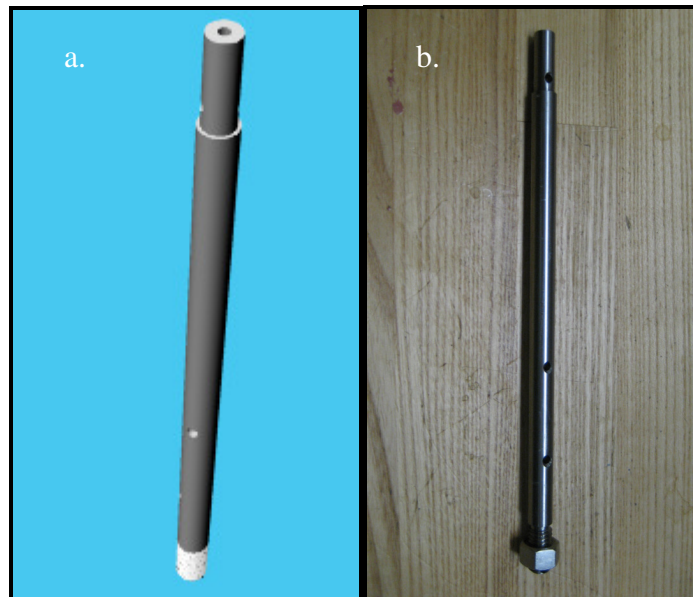
Figure 5-7: Modified thruster shaft design in CAD (a), and machined (b).

The propeller design is such that the blades can be modified and reinserted into the hub without having to manufacture a new hub for the propeller. Additionally, propeller blades can be replaced individually if necessary. The propeller assembly, including the blades, hub, hub insert and thruster shaft is shown in Figure 5-8.

Figure 5-8: Propeller assembly.

The propeller design was validated by comparing the design propeller geometry to the FDM-printed results, as shown in Fig. 5-9. First, the propeller geometry file was exported as a .3dm file. Along each chosen r/R value five evenly-spaced points were chosen. The 3D printed propeller was measured on a Bridgeport milling machine using a dial indicator. Points on the surface of the propeller blade were chosen for three different r/R values, 0.25, 0.70 and 0.80. The propeller was measured in its forward and reverse positions. Figure 5-9 compares the actual blade produced to the desired blade geometry determined in MATLAB.   The measurements of the actual 3D printed propeller can be found in Appendix B.5.   The measurements of the blade geometry at the leading and trailing edge are erroneous due to the difficulty in pinpointing the edge of the blade with a dial indicator and effects due to deflection. However, the other points measured clearly match the desired results, as are geometrically represented in Figure 5-9.

46
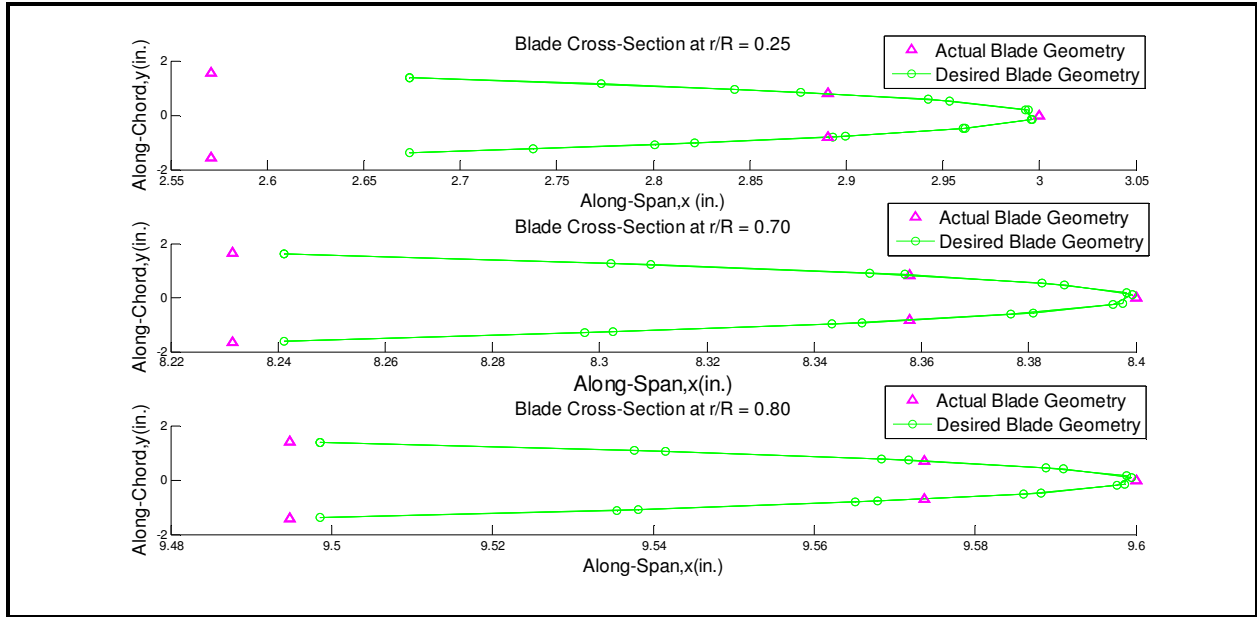
Figure 5-9: Propeller geometry validation, where the actual 3D printed blade is shown in comparison to the desired blade geometry.

5.6 AUV Design Summary

The final optimal propeller was designed with three blades, at 2RPS, a hub starting at r/R(0.2), and a diameter of 0.6096m. A design depth of 2m resulted in no cavitation effects. The design had a modified chord distribution which resulted in a lift coefficient ranging from less than 0.5 to less than 0.24. The initial propeller design had an efficiency of 0.7091, and the final design had an efficiency of 0.7642, for an improvement of 5.51 percent from optimizing the chord distribution.

The existing propeller required attention specifically to the lift coefficient and the value of P/D. The PVL lifting line code does not take into consideration stalling effects of the propeller due to C_L values greater than one. In the case of the off-the-shelf propeller analyzed, stalling effects could occur from the hub to r/R = 0.9. Additionally, the lift coefficient values were considerably higher than that of the design propeller, and likely to cause stalling. In order to complete a rigorous analysis, efficiency losses due to high C_L values should be determined.

The pitch over diameter value calculated for propeller performance was far lower than the actual value taken at r/R = 0.7. Therefore, the efficiency value calculated for the existing

propeller was erroneously dependent on an optimal circulation distribution. The P/D ratio can be increased in future blades by increasing the chordlength or decreasing the propeller diameter.

Under the assumption that the existing propeller is sold off-the-shelf for a scaled range of diameters, analysis showed rewarding performance improvements as a result of increasing *both* the propeller *and* the hub diameters. Increasing the propeller diameter for this propeller's value of N and geometry largely contributed its increase in calculated efficiency. The lift coefficients were within a satisfactory range for an off-the-shelf propeller increased from a 0.594m to 0.76m blade diameter and a hub from r/R = 0.104 to an r/R = 0.31. Furthermore, the efficiency showed a 3.44 percent improvement, from an efficiency of 0.7810 to 0.8154. The efficiency improvement was actually greater than could be measured using the methods in this analysis, since the effect of stalling was not accounted for in computing the efficiency for the original propeller design. The modified propeller's C_L values ranged from0. 6131 at the hub to 0.2839 at the tip. Consideration was made in cutting down the tips of the propeller due to their low chordlengths, but this would be counterproductive due to the fact that the modeled efficiency decreased as propeller diameter decreased, as well as the added effort and cost of modifying the propeller. The optimized design blade did have a lower calculated efficiency, however, the requirements for its range of acceptable C_L values were more stringent, and did not account for loss of efficiency through stalling effects. If the off-the-shelf geometry were held to the range of C_L values less than 0.5, the propeller diameter would get prohibitively large. The original off-the-shelf propeller was unacceptable since it was overloaded, which increased the chances of stalling thus making it inefficient for its application. The performance analysis in this study showed in Table 5-3 that the unmodified off-the-shelf propeller's C_L values were greater than 1.0, and likely to stall.

When using the design codes, the designer must be careful to evaluate the section lift coefficients of the final design; the code does not predict the loss in performance due to stall at high lift coefficients. Also, the user input section drags would need to be increased in high lift ranges. The listing of computed lift coefficients for the blade is provided to the designer to evaluate these effects. The coefficient of lift and pitch over drag make significant contributions, and should not be overlooked.

There are several methods that merit further investigation which would modify the existing propellers in order to provide greater efficiency. First, when a greater propeller diameter can be accommodated, the hub size of the propeller could be increased, which decreases the lift coefficient, and therefore prevents stalling. From basicPVL analysis, propellers with greater diameters show a trend of higher efficiency than other propellers. Under the assumption that the same off-the-shelf propeller geometry exists for greater propeller diameters, investigations showed that the lift coefficient, and thus overall performance, could be improved from both of these modification methods. However, larger propellers would revolve more slowly, and the efficiency of the drive shaft may suffer. Also, practical considerations generally limit the size of a propeller for a given application as well as blade strength issues.

Further investigation into optimal rotational speed will lead to better designs of AUV propellers. In this design, some approximations were made on the range of practical RPS values, due to minimal information on AUV thruster performance. A factor that was not considered, but is essential to overall AUV performance, is the interaction among the AUV's design speed, thruster speed, and power consumption. However, when detailed AUV information is available, the requirements for the propeller design will become more complex, but the same principles for the geometric design remain the same. Therefore, the MATLAB™ code and other methods used for this propeller design provide a useful tool for further applications [15].

6 Case Study: OpenPVL Propeller Prototype Printable File

OpenPVL can be used in a straightforward process to create propellers optimized for a set of necessary design characteristics. In this case, OpenPVL was used to analyze and create the geometry for a model based on the US Navy 4148 propellers. The geometry for the Navy 4148 model is listed in Appendix B.2 [16]. The propeller design was a 1:8 scale model, with characteristics which were determined by using the dimensionless thrust coefficient, $K_T = 0.225$, and the advance coefficient, $J = 0.75$. The model had three blades, and was 0.25 m in diameter. The propeller was designed to run in a tank that could simulate a vehicle speed of 5 m/s, and a thruster that ran at 1600 RPM.

OpenPVL created four propeller design files and two scripting files. As shown in Figure 2-5, the *Single Propeller Design* GUI provides a text input box for the user to create a *Filename Prefix* or to use the default, *OpenPVL*. Upon running the *Single Propeller Design*, the following files were saved:

- *OpenPVL*_Input.txt
- *OpenPVL*_Output.txt
- *OpenPVL*_Performance.txt
- *OpenPVL*_Geometry.dat
- *OpenPVL*_CADfull.txt
- *OpenPVL*_CADblade.txt

6.1 OpenPVL Propeller Design Files

The initial design characteristics were saved in an input file, as in Table 6-1. OpenPVL created files which output a summary of the propeller's design characteristics, the propeller performance, and the propeller geometry. Like MPVL, OpenPVL computed a summary of propeller design characteristics (Table 6-2) at radial increments, in addition to the design thrust coefficient and propeller efficiency, which in this case is 0.6745 [5]. The propeller performance file (Table 6-3) is useful for cavitation analysis and contains a table which lists the total inflow

velocity (V*), the undisturbed flow angle (β), the hydrodynamic pitch angle (βi), the vortex sheet strength (Gamma), the lift coefficient (C_L), the cavitation number (σ), and the pitch angle variation (Δα). The propeller geometry file (Table 6-4) recorded the propeller diameter, number of blades, propeller speed, hub diameter, meanline type, thickness form, pitch, skew, rake, chord distribution, camber distribution and thickness distribution. The propeller geometry files list the blade geometry which can be used to manufacture the propeller.

Table 6-1: OpenPVL Input file.

```
        4148 Sample_Input.txt for Open_PVL_v1

11       Number of Vortex Panels over the Radius
10       Max. Iterations in Wake Alignment
1        Hub Image Flag: 1=YES, 0=NO
1.0      Hub Vortex Radius/Hub Radius
11       Number of Input Radii
3        Number of Blades
0.750    Advance Coef., J, Based on Ship Speed
1.019    Desired Thrust Coef., Ct
0        Hub Unloading Factor: 0=optimum
0        Tip Unloading Factor: 1=Reduced Loading
1        Swirl Cancellation Factor: 1=No Cancellation
r/R       c/D       Cd      Va/Vs  Vt/Vs
0.20000   0.16000   0.00800   1.00  0.0000
0.25000   0.18180   0.00800   1.00  0.0000
0.30000   0.20240   0.00800   1.00  0.0000
0.40000   0.21960   0.00800   1.00  0.0000
0.50000   0.23050   0.00800   1.00  0.0000
0.60000   0.23110   0.00800   1.00  0.0000
0.70000   0.22734   0.00800   1.00  0.0000
0.80000   0.21730   0.00800   1.00  0.0000
0.90000   0.18060   0.00800   1.00  0.0000
0.95000   0.13870   0.00800   1.00  0.0000
1.00000   0.00100   0.00800   1.00  0.0000
```

r/R is the ratio of the radial location, r, to the total length of the blade radius, R.

c/D is the ratio of chordlength to diameter.

Cd is the drag coefficient.

Va/Vs is the axial inflow velocity ratio (is 1, assuming uniform inflow in open water).

Vt/Vs is the tangential inflow velocity ratio (is 0, assuming uniform inflow in open water).

Table 6-2: OpenPVL Output file.

```
              4148 Sample_Output.txt for Open_PVL_v1
                          Output Table
Ct=  1.0186
Cp=  1.5102
Kt=  0.2250
Kq=  0.0398
Va/Vs=  1.0000
Efficiency= 0.6745
 r/R        G        Va       Vt       Ua        Ut       beta     betai     c/D       Cd
0.20407  0.038409  1.00000  0.0000   0.10751  -0.17519  49.476   58.465   0.16182   0.00800
0.23615  0.039456  1.00000  0.0000   0.13174  -0.18552  45.312   54.621   0.17588   0.00800
0.29770  0.043360  1.00000  0.0000   0.17483  -0.19530  38.727   48.165   0.20179   0.00800
0.38374  0.049111  1.00000  0.0000   0.22444  -0.19451  31.886   40.912   0.21744   0.00800
0.48731  0.054296  1.00000  0.0000   0.26813  -0.18298  26.100   34.311   0.23008   0.00800
0.60000  0.056897  1.00000  0.0000   0.30065  -0.16664  21.697   28.998   0.23110   0.00800
0.71269  0.055665  1.00000  0.0000   0.32274  -0.15059  18.519   25.015   0.22658   0.00800
0.81626  0.049952  1.00000  0.0000   0.33706  -0.13732  16.303   22.167   0.21412   0.00800
0.90230  0.039706  1.00000  0.0000   0.34600  -0.12752  14.820   20.232   0.17930   0.00800
0.96385  0.025570  1.00000  0.0000   0.35120  -0.12117  13.911   19.036   0.12112   0.00800
0.99593  0.008828  1.00000  0.0000   0.35358  -0.11807  13.480   18.465   0.04543   0.00800


Ct is the required thrust.
Cp is the power coefficient. Cp = Cq*2*pi/J.
Kt is the thrust coefficient. Kt = Ct*J^2*pi/8, where J is the advance coefficient.
Kq is the torque coefficient. Kq = Cq*J^2*pi/8.
Va/Vs is the axial inflow velocity ratio (is 1, assuming uniform inflow in open water).
Efficiency is the ratio of required thrust times the wake to the power coefficient.
r/R is the ratio of the radial location, r, to the total length of the blade radius, R.
G is the non-dimensional circulation.
Va is the axial inflow velocity.
Vt is the tangential inflow velocity.
Ua us the induced axial velocity.
Ut is the induced tangential velocity.
beta is the flow angle.
betaI is the hydrodynamic Pitch angle.
c/D is the ratio of chordlength to diameter.
Cd is the drag coefficient.
```

Table 6-3: OpenPVL Performance file.

```
         4148 Sample_Performance.txt for Open_PVL_v1
                  Propeller Performance Table
      r/R      V*    beta    betai   Gamma    Cl    sigma   d_alpha
     0.204   4.246   49.48   58.46   0.1508  1.756  14.165   2.20
     0.236   4.893   45.31   54.62   0.1549  1.440  10.663   2.33
     0.298   6.153   38.73   48.17   0.1703  1.097   6.739   2.45
     0.384   7.938   31.89   40.91   0.1929  0.894   4.046   2.44
     0.487  10.103   26.10   34.31   0.2132  0.734   2.495   2.30
     0.600  12.468   21.70   29.00   0.2234  0.620   1.637   2.09
     0.713  14.835   18.52   25.01   0.2186  0.520   1.155   1.89
     0.816  17.011   16.30   22.17   0.1962  0.431   0.877   1.72
     0.902  18.818   14.82   20.23   0.1559  0.370   0.716   1.60
     0.964  20.111   13.91   19.04   0.1004  0.330   0.627   1.52
     0.996  20.785   13.48   18.46   0.0347  0.294   0.587   1.48


     r/R is the ratio of the radial location, r, to the total length of the blade
          radius, R.
     V* is the total inflow velocity.
     beta is the undisturbed flow angle.
     betai is the hydrodynamic Pitch angle.
     Gamma is the vortex sheet strength.
     Cl is the lift coefficient.
     Sigma is the cavitation number.
     d_alpha is the inflow variation bucket width.
```

Table 6-4: OpenPVL Geometry file.

```
          4148 Sample_Geometry.dat for Open_PVL_v1
Propeller Diameter = 0.25 m
Number of Blades = 3
Propeller Speed= 1600 RPM
Propeller Hub Diameter = 0.05 m
Meanline Type: NACA a=0.8
Thickness Type: NACA 65A010


 r/R     P/D     c/D     fo/c    to/c    rake    skew
0.204   1.11    0.162   0.0310  0.2010  0.0     0.000
0.236   1.11    0.176   0.0278  0.1676  0.0     0.000
0.298   1.10    0.202   0.0211  0.1191  0.0     0.000
0.384   1.10    0.217   0.0159  0.0939  0.0     0.000
0.487   1.11    0.230   0.0117  0.0717  0.0     0.000
0.600   1.11    0.231   0.0089  0.0541  0.0     0.000
0.713   1.12    0.227   0.0071  0.0469  0.0     0.000
0.816   1.13    0.214   0.0057  0.0405  0.0     0.000
0.902   1.13    0.179   0.0046  0.0331  0.0     0.000
0.964   1.14    0.121   0.0033  0.0288  0.0     0.000
0.996   1.14    0.045   0.0004  0.0039  0.0     0.000


r/R is the ratio of the radial location, r, to the total length of the blade
 radius, R.
P/D is the ratio of pitch over diameter.
c/D is the ratio of chordlength to diameter.
fo/C is the ratio of camber to chordlength.
to/C is the ratio of thickness to chordlength.
```

6.2 OpenPVL Geometry File into CAD

Once the propeller was designed using OpenPVL, the MATLAB™-scripted command files, *OpenPVL*_CADfull.txt or *OpenPVL*_CADblade.txt could be read such that they produce a full propeller with the desired number of blades and a propeller blade only, respectively. The command file for the full propeller produces a blade based on the value of the height variable, and accommodating room for a fillet. In the 3D-printable design, the blades extend to 95 percent of the design hub radius, to ensure that the blade can be joined to the hub when the command file is run. An example of a RHINO™-compatible script for the Navy 4148 based model propeller detailed in this chapter can be found in Appendix C. Depending on user propeller needs, the blade may be artificially thickened. Also, it is simple to add desired propeller features, such as fillets at the hub, and fairings, slots for pin connections, etc.

The MATLAB™ scripting file created by OpenPVL was made specifically to provide a RHINO™ interface, although future versions of OpenPVL will be compatible with other CAD programs, such as SolidWorks™. The following are several techniques used to produce the propeller, as shown in Fig. 6-2, which was attached to a hub and filleted in RHINO™:

- OpenPVL chooses to extend the blade into the hub by plotting x, y and z dimensional values which start at 95 percent of the hub's radius and extend to the propeller's tip. Extending the blade into the hub geometry ensures that the blade and hub will be able to be joined using a Boolean Union.

- If a *Boolean Union* fails, consider the direction of the surface normals for the two objects you are trying to unify, using *Dir*.

- To create a propeller fillet, several aspects must be considered. First, the height of the hub must accommodate the fillet's radius. Second, the blade must not have a trailing edge value of zero. This creates a sharp edge, which is incompatible with the *fillet* command. Next, when using *FilletSrf* to fillet the hub to the blade surface, the fillet generally works the best when the surface normal direction is outward for both the hub and the blade.

- A propeller fairing can be built into the hub by using the *paraboloid* function.

- Creating a .STL file in RHINO™ requires the following procedure:

  1) After all custom propeller modifications have been made, apply the command *Join* to all the polysurfaces. For closed surfaces, use a *Boolean Union.*

  2) Once all the surfaces are joined, use *Mesh*. Ensure that the mesh settings are such that the minimum edge length exceeds the size of the strand of plastic used to fabricate the propeller. For example, the minimum edge length was set to 0.025m when printed using an FDM strand size of 0.007m. For best results, check the boxes for *refine mesh* and *pack textures*.

  3) Next, delete all of the polysurfaces to make sure the mesh has been completed for the desired portions of the propeller.

  4) Use the *Weld* command. Select the polygon mesh surface, and choose and angle tolerance of 180 degrees.

  5) Next, use the *UnifyMeshNormals* command. This ensures that the surface normals for the mesh object are all going in the same direction, i.e., inward or outward.

  6) Finally, select *SelNakedMeshEdgePt* to determine that all of design is *watertight.* A file must be considered *watertight* to be saved as a .STL file.
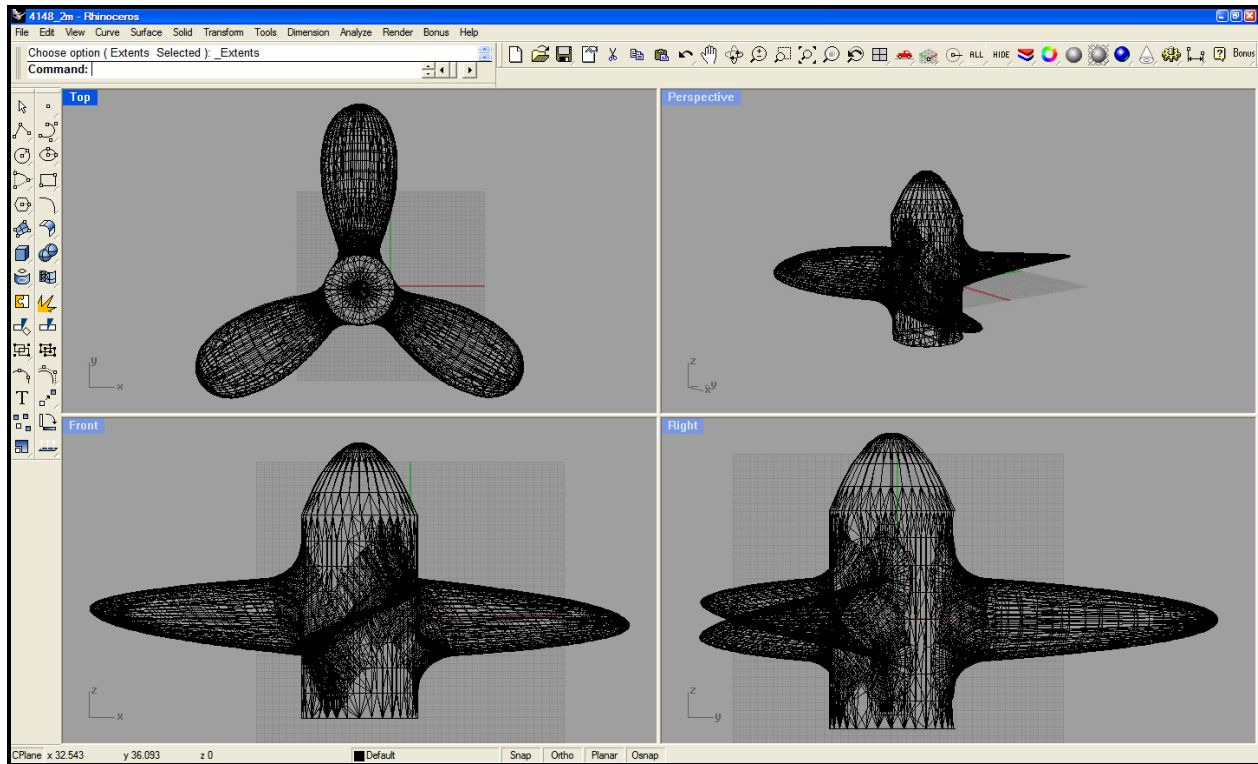
Figure 6-1: Using RHINO™ to create a .STL file with the 4148-based model propeller.

The printable file made using the 4148-based test case geometry is shown in Figure 6-3.
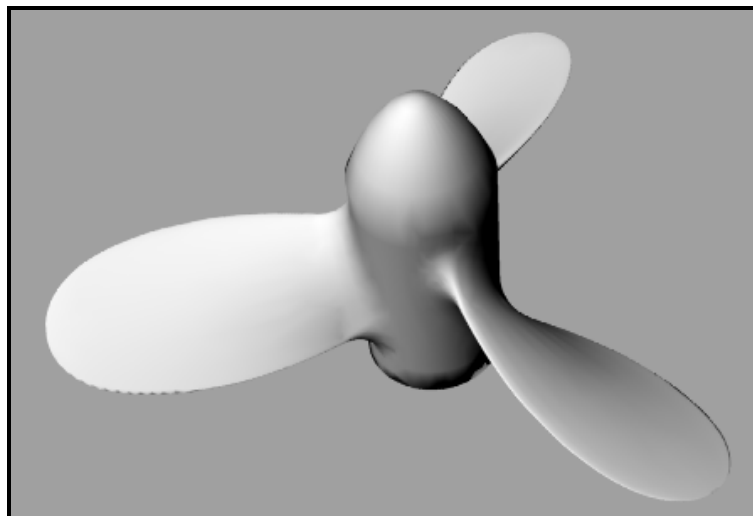
Figure 6-2: Sample propeller .STL file using OpenPVL 4148 geometry.

This study followed the necessary steps for propeller prototyping as outlined in Fig. 3-1. The OpenPVL files can be saved under a name of the user's choosing. This allows the user to create, compare, and store designs for several different propeller blades. The test case revealed the files which OpenPVL produces for analysis purposes, and showed that the geometry based on the US Navy 4148 propeller could be modeled with these techniques. OpenPVL's versatility allows it to efficiently produce a wide-range of propellers, not limited to AUVs, using this method.

7 Conclusions and Future Studies

7.1 Conclusions

OpenPVL has been developed from MPVL [5] to support a propeller design-to-prototype process. MPVL was modified to include 3D printing capabilities, as well as to save the propeller design characteristics into organized files for each test run. Lift coefficient analysis and initial blade parameterization was achieved by using basicPLV. The OpenPVL code was exercised in various parts of this design work, as an interface between the MATLAB™ propeller geometry outputs and the CAD software, RHINO™. Additionally, OpenPVL was used to make performance comparisons among different propeller designs. Finally, this study has provided an outline for propeller design and prototyping by describing techniques employed to design in OpenPVL, how create a custom propeller and hub using CAD software, and produce a propeller using an FDM.

An initial design code, basicPVL, allows users to produce a script from which a CAD program can read the propeller design geometry. The PVL-based code generates a series of contour graphs to help the user determine a starting point for the propeller design geometry. However, it is not very accessible to new users, and has limited design inputs. The comparison of an existing design to an optimized design using propeller lifting line code improved existing propellers and contributed to the process of choosing future off-the-shelf propellers.

OpenPVL guides the user through the process of designing an efficient propeller and creating propeller geometry files which interface with a CAD program. The CAD program allows the user to create a 3D-printable file and to add a custom propeller hub, fillets and a fairing, if desired. Investigations into the 3D printing process revealed that ABS-PC is the best choice of printing material in an FDM machine. Casting a mold from a propeller prototype would afford additional freedom of propeller material choices and manufacturing capabilities.

The test case of an off-the-shelf propeller used the Bollard Test and the Moving Cart test to investigate propeller thrust performance. The test results showed that the performance of off-

the-shelf propellers was limited, in terms of producing vehicle speed and power consumption. This substantiates the need for software capable of custom-designed propeller blades. The experiment would be well-supplemented by experimenting with additional motor gear boxes, to allow for a different range of rotational speeds tested.

Initial design decisions from basicPVL code for the case study of an AUV chose a three-bladed 0.625m diameter propeller at 2RPS, with a hub located at r/R(0.2) and an efficiency of 0.7091. Final propeller design modified the chord distribution to improve distributed lift coefficients and to ensure that shallow-water operations would not experience cavitation effects. The final design geometry used the same values of the propeller blade number, diameter, RPS and hub location as the initial design, but at an efficiency of 0.7642. While off-the-shelf propellers, such as those designed for model airplanes, can have high calculated efficiencies relative to actuator disk theory, the lift coefficient at each radial position and the resulting stalling effects must be considered.

OpenPVL demonstrated the ability to perform an analysis for the custom-designing and rapid-prototyping of propellers. *Single Propeller Design*, an easy-to-use and modify GUI in the OpenProp design suite, uses OpenPVL to create two propeller design files from which propellers may be printed, either as a complete propeller and hub, or as separate blades, for larger propeller designs. The files can be used for direct comparison among multiple designs. In this way, OpenPVL promotes a rapid transition between the propeller design and the optimized prototype.

7.2 Developmental Goals for OpenPVL

OpenPVL was initiated by the opportunity to add design features to MPVL [5]. Likewise, additional tools will be added to the OpenProp software suite. Several plans for future development include:

- Propeller motor-matching techniques.
- Propeller hub and fairing designs.
- An iterative optimization process for chordlength, the pitch to diameter ratio, and the camber and thickness ratio.

58

- Cavitation number calculation through automated Brockett diagrams.
- Ducted propeller capabilities.
- Ability to interface with CAD programs such as SolidWorks™, in addition to RHINO™.
- Finite Element Analysis (FEA)

Most significantly, OpenPVL is readily accessible and modifiable by all propeller users; it provides a basic platform which will easily incorporate the addition of advanced codes and design tools. This propeller design tool is ready for immediate use in streamlined propeller prototyping and also holds many opportunities for future expansion and development.

References

[1] M. Purcell, C. von Alt, B. Allen, T. Austin, N. Forrester, et al., "New capabilities of the REMUS autonomous underwater vehicle," *OCEANS 2000 MTS/IEEE Conference and Exhibition*, vol. 1, pp. 147-151, September, 2000.

[2] "Timetable: Development of the Propeller," 2002. http://www.mh-aerotools.de/airfoils/prophist.htm. Retrieved May 13, 2006.

[3] Carlton, J. S. *Marine Propellers and Propulsion.* Oxford: Butterworth-Heinemann Ltd., 1994.

[4] J. E. Kerwin. Hydrofoils and Propellers. Cambridge, MIT CopyTech, 2001.

[5] H. Chung, "An Enhanced Propeller Design Program Based on Propeller Vortex Lattice Lifting Line Theory," Masters Thesis, Massachusetts Institute of Technology, June 2007.

[6] Coney, William B., Ching-Yeh Hsin, and Justin E. Kerwin. *MIT-PLL-2 Report and User's Manual.* Cambridge,MA: Massachusetts Institute of Technology, 1986.

[7] R. Kimball, "Huang Body-1 MTWAKE screen matching experimental wake MTSOL grid finer near body," personal communication, April 19, 2006.

[8] Abbott, Ira H., and Albert E. Von Doenhoff. *Theory of Wing Sections.* New York: Dover Publications, 1959.

[9] T. Brockett, "Minimum Pressure Envelopes for Modified NACA-66 Sections with NACA a-0.8 Camber and BuShips Type I and Type II Sections," DTMB Report 1780, February 1966.

[10] J. N. Newman, *Marine Hydrodynamics.* Cambridge, MA: The MIT Press, 1977, pp. 206-208.

[11] "Titan FDM Real Materials." http://www.StrataSys™.com. Accessed 05JUL2007.

[12] "Machinist-Materials, Plastics Comparison Table." http://www.machinist-materials.com/comparison_table_for_plastics.htm. Accessed 03JUL2007.

[13] R. Camilli, "Cosmos FloWorks™ AUV Modeling," personal communication, April 21, 2006.

[14] C. Roman, "SeaBED MATLAB™ Code," personal communication, April 3, 2006.

[15] K. P. D'Epagnier, "AUV Propellers: Optimal Design and Improving Existing Propellers for Greater Efficiency," in Proc. OCEANS, Boston, MA, 2006.

[16] Dr. S. A. Kinnas, "University/Navy/Industry Consortium on Cavitation of High Speed Propulsors," fifth meeting, June 1st and 2nd, 1995.

Appendix A: OpenPVL MATLAB™ Source Code

```matlab
% OpenProp Version 1.0 ======= Last modified: 18AUG2007 by K. P. D'Epagnier
% ========================================================================
% Copyright 2007 Hsin-Lung Chung; Kathryn D'Epagnier
% This program is free software; you can redistribute it and/or modify it
% under the terms of the GNU General Public License version 2
% as published by the Free Software Foundation.
% This prorgam is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
% See the GNU General Public License for more details.
% Combined with RHINO™ scripting 03MAY2007 by Kathryn D'Epagnier.
% ========================================================================
function OpenPVL_v1(action)
            %outputs we want:
KRC(i),pitchOD(i),Rake(i),Skew(i),kCDC(i),F0oc(i),T0oc(i).*kCDC(i));
            %sent to RHINO™filemaker named as:
            %[r_R,P_D,rake,deg_skew,C_D,F0_C,T0_D]
if nargin==0                          % nargin is "Number of Input Arguments"
   action = 'Initiate_Fig';
end
% ----------------------------------------------- Set up global variables
global Fig_Main Parametric Single Parametric_Flag Single_Flag XR0 XCHD_in XCD_in XVA_in ...
      XVT_in Text1_P Text2_P Text3_P NBLADE_in N_in D_in Input1 Hub_Flag Input1_S ...
      Input2_S Input3_S Text1_S Text2_S Text3_S f0oc_in t0oc_in skew_in rake_in ...
      Run_Parametric Run_Single Fig_P Fig1_S Fig2_S Fig3_S New_Parametric New_Single ...
      Err_Parametric Err_Single Label_Col Label_Row Welcome lXRO D NBLADE Text4_S Input4_S ...
      filenamek xur xlr yur ylr zlr zur f t x_sec_X x_sec_Y LocalChord RC section Rhub Msg_P
Dhub D tfactor;
% =================================================== INITIATE MAIN FIGURE
if strcmp(action,'Initiate_Fig')==1
    % ============================== Declare Default Input Variables
    % Common_def: thrust,Vs,Dhub,MT,ITER,RHV,NX,HR,HT,CRP,rho
    Common_Def=[625 5 .2 25 10 1 11 0 0 1 1000];
    XR0=[.2 .25 .3 .4 .5 .6 .8 .9 .95 1];                    % r/R
    %XR0=[.104 .160 .216 .328 .440 .552 .776 .888 .944 1]; %off-the-shelf hub .664
    %XRO=[.2 .3 .4 .5 .6 .7 .8 .9 .95 1];%hub .2    .7

    %XRO=[.104 .160 .216 .328 .440 .552 .664  .888 .944 1.0];% r/R for off-the-shelf 0.7760
    %XCHD_def=[.1600 .1818 .2024 .2196 .2305 .2311 .2173 .1806 .1387 .0010];  % c/D

    XCHD_def =[0.2269  0.2193    0.2148    0.2029    0.1885    0.1693    0.1545    0.1275
0.0801    0.0572    0.0032];%c/D for printed prop
    %XCHD_def=[0.0770 0.0742 0.0751 0.0846 0.0814 0.0739 0.0533 0.0386 0.0301 0.0017];%c/D off-
the-shelf 0.0647
    % XCHD_def=[0.0745 0.0772 0.0819 0.0840 0.0779 0.0702  0.0505 0.0369 0.0288 0.0017];% hub
.20.0611
    XCD_def=ones(length(XR0)).*.008;                        % Cd
    XVA_def=ones(length(XR0));                              % Va/Vs
    XVT_def=zeros(length(XR0));                             % Vt/Vs
    f0oc_def=[.0174 .0195 .0192 .0175 .0158 .0143 .0133 .0125 .0115 .0000]; % f0/c
    t0oc_def=[.2056 .1551 .1181 .0902 .0694 .0541 .0419 .0332 .0324 .0000]; % t0/c
    skew_def=zeros(length(XR0));                            % Skew (degrees)
    rake_def=zeros(length(XR0));                            % Rake (Xs/D)
    Parametric_def = [3 6 1 50 200 50 .5 5 .5];              % Z,N,D (Min,Max,Increment)
    Single_def1=[3 1600 .25];                               % NBLADE, N and D
    Single_def2=[3 .3 1.54 20];                             % H, dV, AlphaI and NP
    Single_def3={'OpenPVL'};                    % filenamek
    % ------------------ Parameters that define the layout of UI components
    ew = .06;    eh = .04;   tw1 = .15;  tw2 = .35;   ph = .06;   X1 = .02;     X2 = .085;
    X3 = .45;    Y1 = .93;   Y2 = .7;    Y3 = .18;    Y4 = .015;  h1 = eh+.005;
    % ----------------------------------------------- Create Figure for GUI
    close all;
    Fig_Main=figure('units','normalized','position',[.02 .3 .96 .6],...
        'numbertitle','off','name','OpenPVL_v1','menubar','none');
    Frame=uicontrol('parent',Fig_Main,'style','frame',...
        'units','normalized','position',[0 0 1 1]);
    % ----------------------------------------- Create UI menu components
    Menu=uimenu(Fig_Main,'label','Options');
    Parametric=uimenu(Menu,'label','Parametric Analysis','callback',...
```

```matlab
        'OpenPVL_v1(''Initiate_Parametric'')');
Single=uimenu(Menu,'label','Single Propeller Design','callback',...
    'OpenPVL_v1(''Initiate_Single'')');
uimenu(Menu,'label','Print Screen','separator','on','callback',['printpreview']);
uimenu(Menu,'label','Exit','separator','on','callback',['clear all; close all; clc']);
Parametric_Flag = 0;      Single_Flag = 0;    % Flags thet indicate the status of program
% ------------------------------------------------------ Welcome message
Welcome=uicontrol('style','text','units','normalized','FontSize',14,...
    'FontWeight','bold','position',[.2 .4 .6 .2],'string',...
    'To Start, Select an Action in the Option Menu.');
% ------------------------------------------------ Create Text Labels
Label1_P={'Min' 'Max' 'Increment'};
Label2_P={'Number of blades' 'Propeller speed (RPM)' 'Propeller diameter (m)'};
Label3_P={'Required Thrust (N)'                          % THRUST
          'Ship Velocity (m/s)'                          % V
          'Hub Diameter/Diameter Ratio'                  % Dhub
          'Num. Vortex Panels over Radius'               % MT
          'Max. Iterations in Wake Alignment'            % ITER
          'Hub Vortex Radius/Hub Radius'                 % RHV
          'Number of Input Radii'                        % NX
          'Hub Unloading Factor: 0=Optimum'              % HR
          'Tip Unloading Factor: 1=Reduced Loading'      % HT
          'Swirl Cancellation Factor: 1=No Cancellation' % CRP
          'Water Density (kg/m^3)'};                     % rho
Label1_S={'Number of Blades'                             % NBLADE
          'Propeller Speed (RPM)'                         % N
          'Propeller Diameter (m)'};                      % D
Label2_S={'Shaft Centerline Depth (m)'                   % H
          'Inflow Variation (m/s)'                        % dV
          'Ideal Angle of Attack (degrees)'               % AlphaI
          'Number of Points over the Chord'};             % NP
Label3_S=['Filename Prefix'];                            % filenamek
% -------------------- Create UI components for Parametric Analysis GUI
for i=1:length(Label1_P)
    NBLADE_in(i)=uicontrol('style','edit','units','normalized','FontSize',10,...
        'FontWeight','bold','backgroundcolor','w','position',[tw1+ew*(i) Y1-h1 ew eh],...
        'string',Parametric_def(i),'callback','OpenPVL_v1(''Update'')','visible','off');
    N_in(i)=uicontrol('style','edit','units','normalized','FontSize',10,'FontWeight',...
        'bold','backgroundcolor','w','string',Parametric_def(i+3),'position',...
        [tw1+ew*(i) Y1-h1*2 ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
    D_in(i)=uicontrol('style','edit','units','normalized','FontSize',10,'FontWeight',...
        'bold','backgroundcolor','w','position',[tw1+ew*(i) Y1-h1*3 ew eh],...
        'string',Parametric_def(i+6),'callback','OpenPVL_v1(''Update'')','visible','off');
    Text1_P(i)=uicontrol('style','text','units','normalized','FontSize',9,...
        'FontWeight','bold','position',[tw1+ew*(i) Y1 ew eh],'string',Label1_P(i),...
        'visible','off');
    Text2_P(i)=uicontrol('style','text','units','normalized','FontSize',10,...
        'FontWeight','bold','HorizontalAlignment','left','visible','off',...
        'position',[X1 Y1-h1*(i) tw1 eh],'string',Label2_P(i));
end
% ------------------------- Create UI components for Single Design GUI
for i = 1:length(Single_def1)
    Input1_S(i)=uicontrol('style','edit','units','normalized','FontSize',10,...
        'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')',...
        'position',[X1 Y1-h1*(i-1) ew eh],'string',Single_def1(i),'visible','off');
    Text1_S(i)=uicontrol('style','text','units','normalized','FontSize',10,...
        'FontWeight','bold','HorizontalAlignment','left','position',...
        [X2 Y1-h1*(i-1) tw2 eh],'string',Label1_S(i),'visible','off');
end
for i = 1:length(Single_def2)
    Input2_S(i)=uicontrol('style','edit','units','normalized','FontSize',10,...
        'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')',...
        'position',[X1 Y3-h1*(i-1) ew eh],'string',Single_def2(i),'visible','off');
    Text2_S(i)=uicontrol('style','text','units','normalized','FontSize',10,...
        'FontWeight','bold','HorizontalAlignment','left','visible','off',...
        'position',[X2 Y3-h1*(i-1) tw2 eh],'string',Label2_S(i));
end
Input4_S=uicontrol('style','edit','units','normalized','FontSize',10,...
        'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')',...
```

```matlab
            'position',[X3 Y3-eh 3*ew eh],'string',Single_def3,'visible','off');
        Text4_S=uicontrol('style','text','units','normalized','FontSize',10,...
            'FontWeight','bold','HorizontalAlignment','left','visible','off',...
            'position',[X3 Y3 tw2 eh],'string',Label3_S);
    % --------------------------- Create UI components for Common Inputs
    for i = 1:length(Common_Def)
        Input1(i)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')',...
            'position',[X1 Y2-h1*(i-1) ew eh],'string',Common_Def(i),'visible','off');
        Text3_P(i)=uicontrol('style','text','units','normalized','FontSize',10,...
            'FontWeight','bold','HorizontalAlignment','left','visible','off',...
            'position',[X2 Y2-h1*(i-1) tw2 eh],'string',Label3_P(i));
    end
    Hub_Flag=uicontrol('style','checkbox','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3 Y1 tw2 eh],'value',1,'callback',...
        'OpenPVL_v1(''Update'')','string','Hub Image Flag (Check for YES)','visible','off');
    % ------------------------------------------------- Create POP-UP MENU
    Input3_S(1)=uicontrol('style','popupmenu','units','normalized','FontSize',8,...

'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')','position',...
        [X3 Y1-eh*3 tw1 eh],'string',{'NACA a=0.8' 'Parabolic'},'visible','off');
    Text3_S(1)=uicontrol('style','text','units','normalized','FontSize',10,...
        'FontWeight','bold','HorizontalAlignment','left','position',[X3 Y1-eh*2 tw1 eh],...
        'string','Meanline Type:','visible','off');
    Input3_S(2)=uicontrol('style','popupmenu','units','normalized','FontSize',8,...

'FontWeight','bold','backgroundcolor','w','callback','OpenPVL_v1(''Update'')','position',...
        [X3+tw1+ew Y1-eh*3 tw1 eh],'string',{'NACA 65A010' 'Elliptical' 'Parabolic' 'NACA
Thickened for 3-D Printing'},...
        'visible','off');
    Text3_S(2)=uicontrol('style','text','units','normalized','FontSize',10,'FontWeight',...
        'bold','HorizontalAlignment','left','position',[X3+tw1+ew Y1-eh*2 tw1 eh],...
        'string','Thickness Form:','visible','off');
    % ------------- Create UI components for inputs on the right of the GUI
    for j = 1:length(XR0)
        Label_Row(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','position',[X3 Y2-h1*j ew eh],'string',XR0(j),...
            'visible','off','enable','inactive');
        XCHD_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',XCHD_def(j),'position',...
            [X3+ew Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        XCD_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',XCD_def(j),'position',...
            [X3+ew*2 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        XVA_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',XVA_def(j),'position',...
            [X3+ew*3 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        XVT_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',XVT_def(j),'position',...
            [X3+ew*4 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        f0oc_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',f0oc_def(j),'position',...
            [X3+ew*5 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        t0oc_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',t0oc_def(j),'position',...
            [X3+ew*6 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        skew_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',skew_def(j),'position',...
            [X3+ew*7 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        rake_in(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',rake_def(j),'position',...
            [X3+ew*8 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
        Msg_P(j)=uicontrol('style','edit','units','normalized','FontSize',10,...
            'FontWeight','bold','backgroundcolor','w','string',rake_def(j),'position',...
            [X3+ew*9 Y2-h1*j ew eh],'callback','OpenPVL_v1(''Update'')','visible','off');
    end
    ColName={'r/R' 'c/D' 'Cd' 'Va/Vs' 'Vt/Vs' 'f0/c' 't0/c' 'Skew' 'Xs/D'};
    for i = 1:length(ColName)
        Label_Col(i)=uicontrol('style','edit','units','normalized','FontSize',10,...
```

```matlab
            'FontWeight','bold','position',[X3+ew*(i-1) Y2 ew eh],'string',ColName(i),...
            'enable','inactive','visible','off');
    end
    % ------------------------------------------------- Create Pushbuttons
    Run_Parametric = uicontrol('style','pushbutton','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3 Y4 ph*2 ph],'string','Run OpenPVL_v1','callback',...
        'OpenPVL_v1(''Execute_Parametric'')','visible','off');
    Run_Single = uicontrol('style','pushbutton','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3 Y4 ph*2 ph],'string','Run OpenPVL_v1','callback',...
        'OpenPVL_v1(''Execute_Single'')','visible','off');
    New_Parametric = uicontrol('style','pushbutton','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3+ph*2 Y4 ph*2 ph],'string','Try Again',...
        'callback','OpenPVL_v1(''Update'')','visible','off');
    New_Single = uicontrol('style','pushbutton','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3+ph*2 Y4 ph*2 ph],'string','Try Again',...
        'callback','OpenPVL_v1(''Update'')','visible','off');
    Err_Parametric = uicontrol('style','pushbutton','units','normalized','FontSize',8,...
        'Fontweight','bold','position',[X3+ph*4 Y4 ph*3 ph],'Foregroundcolor','r',...
        'callback','OpenPVL_v1(''Update'')','visible','off');
    Err_Single = uicontrol('style','pushbutton','units','normalized','FontSize',8,...
        'Fontweight','bold','position',[X3+ph*4 Y4 ph*3 ph],'Foregroundcolor','r',...
        'callback','OpenPVL_v1(''Update'')','visible','off');
    Msg_P = uicontrol('style','text','units','normalized','FontSize',10,...
        'FontWeight','bold','position',[X3 Y3-eh .5 eh*2],'visible','off','string',...
        '* The chord distribution (c/D) is based on the minimum number of blades.  As the number
of blades increases, the chord lengths are scaled to maintain a constant total blade area.',...
        'HorizontalAlignment','left');
% ===================================== INITIATE PARAMETRIC ANALYSIS GUI
elseif strcmp(action,'Initiate_Parametric')==1
    set(Welcome,'visible','off');               set(Parametric,'enable','off');
    set(Single,'enable','on');                  Parametric_Flag = 1;
    if Single_Flag ==1          % If Single Design GUI is already launched
        set(Input1_S,'visible','off');          set(Text1_S,'visible','off');
        set(Input2_S,'visible','off');          set(Text2_S,'visible','off');
        set(Input3_S,'visible','off');          set(Text3_S,'visible','off');
        set(f0oc_in,'visible','off');           set(t0oc_in,'visible','off');
        set(skew_in,'visible','off');           set(rake_in,'visible','off');
        set(Run_Single,'visible','off');        set(New_Single,'visible','off');
        set(Err_Single,'visible','off');
        set(Input4_S,'visible','off');          set(Text4_S,'visible','off');
    end
    set(Text1_P,'visible','on');
    set(Text2_P,'visible','on');
    set(NBLADE_in,'enable','on','visible','on');
    set(NBLADE_in(3),'Enable','off');
    set(N_in,'enable','on','visible','on');
    set(D_in,'enable','on','visible','on');
    set(Input1,'enable','on','visible','on');
    set(Text3_P,'visible','on');
    set(Hub_Flag,'enable','on','visible','on');
    set(Label_Col(1:5),'visible','on');
    set(Label_Col(6:9),'visible','off');
    set(Label_Row,'visible','on');
    set(XCHD_in,'enable','on','visible','on');
%set(XCHD_in(lXRO),'enable','off','visible','on');
    set(XCD_in,'enable','on','visible','on');       set (Msg_P, 'visible','on');
    set(XVA_in,'enable','on','visible','on');
    set(XVT_in,'enable','on','visible','on');
    set(Run_Parametric,'enable','on','visible','on');
    Single_Flag = 0;
% ================================================= UPDATE INPUTS FIELDS
elseif strcmp(action,'Update')==1
    % --------------------------------------- Update Common Input Fields
    set(Input1,'enable','on');                  set(Hub_Flag,'enable','on');
    set(XCHD_in,'enable','on');                 set(XCD_in,'enable','on');
    set(XVA_in,'enable','on');                  set(XVT_in,'enable','on');
    get(Input1,'string');                       get(Hub_Flag,'value');
    get(XCHD_in,'string');                      get(XCD_in,'string');
    get(XVA_in,'string');                       get(XVT_in,'string');
```

```matlab
        % ------------------------------------- Update Respective Input Fields
    if Parametric_Flag==1
        set(NBLADE_in(1:2),'enable','on');       set(N_in,'enable','on');
        set(D_in,'enable','on');                 set(Run_Parametric,'enable','on');
        set(New_Parametric,'visible','off');     set(Err_Parametric,'visible','off');
        get(NBLADE_in,'string');                 get(N_in,'string');
        get(D_in,'string');
    elseif Single_Flag==1
        set(Run_Single,'enable','on');           set(Input1_S,'enable','on');
        set(Input2_S,'enable','on');             set(Input3_S,'enable','on');
        set(f0oc_in,'enable','on');              set(t0oc_in,'enable','on');
        set(skew_in,'enable','on');              set(rake_in,'enable','on');
        set(New_Single,'visible','off');         set(Err_Single,'visible','off');
set(Input4_S,'enable','on');
        get(Input1_S,'string');                  get(Input2_S,'string');
        get(Input3_S,'string');                  get(f0oc_in,'string');
        get(t0oc_in,'string');                   get(skew_in,'string');
        get(rake_in,'string');                   get(Input4_S,'string');
    end
% ================================== PERFORM PARAMETRIC ANALYSIS ALGORITHM
elseif strcmp(action,'Execute_Parametric')==1
    tic;        % start stopwatch
    % ------- Disable components to prevent intervention during calculation
    set(NBLADE_in,'enable','off');               set(N_in,'enable','off');
    set(D_in,'enable','off');                    set(Input1,'enable','off');
    set(Hub_Flag,'enable','off');                set(XCHD_in,'enable','off');
    set(XCD_in,'enable','off');                  set(XVA_in,'enable','off');
    set(XVT_in,'enable','off');                  set(Run_Parametric,'enable','off');
    set(New_Parametric,'visible','on','enable','on');
    % ----------------------------------- Close figures previously created
    if ishandle(Fig_P)~=0
        close(Fig_P);
    end
    if ishandle(Fig1_S)~=0
        close(Fig1_S);
    end
    if ishandle(Fig2_S)~=0
        close(Fig2_S);
    end
    if ishandle(Fig3_S)~=0
        close(Fig3_S);
    end
    % ----------------------------------------------------- Define Variables
    NBLADE=str2double(get(NBLADE_in(1),'string')):1:str2double(get(NBLADE_in(2),'string'));
    N=str2double(get(N_in(1),'string')):str2double(get(N_in(3),'string')):...
        str2double(get(N_in(2),'string'));
    D=str2double(get(D_in(1),'string')):str2double(get(D_in(3),'string')):...
        str2double(get(D_in(2),'string'));
    % --------------------------- Set constraints on propeller parameters
    if (max(NBLADE>6)) || (min(NBLADE)<2)
        set(Err_Parametric,'visible','on','enable','on','string','2<=Number of Blades=<6');
        set(New_Parametric,'enable','off');         beep;             return;
    elseif (str2double(get(N_in(1),'string'))>str2double(get(N_in(2),'string')))
        set(Err_Parametric,'visible','on','enable','on','string','Check Propeller Speed');
        set(New_Parametric,'enable','off');         beep;             return;
    elseif (str2double(get(D_in(1),'string'))>str2double(get(D_in(2),'string')))
        set(Err_Parametric,'visible','on','enable','on','string','Check Propeller Diameter');
        set(New_Parametric,'enable','off');         beep;             return;
    end
    % --------------------------- Derive Common Input Fields in a String
    string = str2double(get(Input1,'string'));
    THRUST = string(1);           V = string(2);              Dhub = string(3).*D;
    MT = string(4);               ITER = string(5);           RHV = string(6);
    NX = string(7);               HR = string(8);             HT = string(9);
    CRP = string(10);             rho = string(11);           Rhub = Dhub/2;
    IHUB = get(Hub_Flag,'value');
    XCHD0 = str2double(get(XCHD_in,'string'));
    XCD0 = str2double(get(XCD_in,'string'));
    XVA0 = str2double(get(XVA_in,'string'));
```

```matlab
        XVT0 = str2double(get(XVT_in,'string'));
        %------------------------------------------------- Perform Algorithm
        for k=1:length(NBLADE)
            for i=1:length(N)
                n(i) = N(i)/60;
                for j=1:length(D)
                    R(j) = D(j)/2;
                    ADVCO = V/(n(i)*D(j));
                    CTDES = THRUST/(rho*V^2*pi*R(j)^2/2);
                    XR1 = Rhub(j)/R(j):(1-Rhub(j)/R(j))/(NX-3):1; %changed 22MAY07
                    half1 = (XR1(1)+XR1(2))/2;
                    half2 = (XR1(NX-3)+XR1(NX-2))/2;
                    XR = cat(2,XR1(1), half1, XR1(2:NX-3), half2, XR1(NX-2));
                    XCHD = pchip(XR0, XCHD0, XR).*(NBLADE(1)/NBLADE(k));    %change blade area to
scale with the number of blades desired
                    XCD = pchip(XR0,XCD0,XR);
                    XVA = pchip(XR0,XVA0,XR);
                    XVT = pchip(XR0,XVT0,XR);
                    % ==================================== Call Main Function
                    [CT,CP,KT,KQ,WAKE,EFFY0,RC,G,VAC,VTC,UASTAR,UTSTAR,TANBC,TANBIC,CDC,CD,KTRY]...
                    =Main(MT,ITER,IHUB,RHV,NX,NBLADE(k),ADVCO,CTDES,HR,HT,CRP,XR,XCHD,XCD,XVA,XVT);
                    % =======================================================
                    EFFY(i,j,k) = EFFY0(end);
                end
            end
        end
        Fig_P=figure('units','normalized','position',[0.01 .06 .4 .3],'name','Efficiency',...
            'numbertitle','off');
        for i = 1:length(NBLADE)
            if length(NBLADE)==4
                subplot(2,2,i);
            else
                subplot(length(NBLADE),1,i);
            end
            plot(D,EFFY(:,:,i));
            str_suffix={' RPM'};
            for j=1:length(N)
                str_legend(j)=strcat(num2str(N(j)),str_suffix);
            end
            legend(str_legend,'location','southwest');          grid on;
            xlabel('Propeller Diameter (m)');                   ylabel('Efficiency');
            title_prefix = {'Number of Blades: '};
            title(strcat(title_prefix,num2str(NBLADE(i))))
        end
        set(New_Parametric,'enable','on');
        figure(Fig_Main);
        toc      % Stop stopwatch
% ============================================ INITIATE SINGLE DESIGN GUI
elseif strcmp(action,'Initiate_Single')==1
    set(Welcome,'visible','off');                   set(Parametric,'enable','on');
    set(Single,'enable','off');                     Single_Flag = 1;
    if Parametric_Flag ==1  % If Parametric Analysis GUI is already launched
        set(Text1_P,'visible','off');               set(Text2_P,'visible','off');
        set(NBLADE_in,'visible','off');             set(N_in,'visible','off');
        set(D_in,'visible','off');                  set(Run_Parametric,'visible','off');
        set(New_Parametric,'visible','off');        set(Err_Parametric,'visible','off');
        set(Msg_P,'visible','off');
    end
    set(Input1_S,'visible','on','enable','on');     set(Text1_S,'visible','on');
    set(Input1,'visible','on','enable','on');       set(Text3_P,'visible','on');
    set(Input2_S,'visible','on','enable','on');     set(Text2_S,'visible','on');
    set(Hub_Flag,'visible','on','enable','on');
    set(Input3_S,'visible','on','enable','on');     set(Text3_S,'visible','on');
    set(Label_Col,'visible','on');                  set(Label_Row,'visible','on');
    set(XCHD_in,'visible','on','enable','on');      %set(XCHD_in,'visible','on','enable','off');
    set(XCD_in,'visible','on','enable','on');
    set(XVA_in,'visible','on','enable','on');       set(XVT_in,'visible','on','enable','on');
    set(f0oc_in,'visible','on','enable','on');      set(t0oc_in,'visible','on','enable','on');
    set(skew_in,'visible','on','enable','on');      set(rake_in,'visible','on','enable','on');
```

```matlab
    set(Run_Single,'visible','on','enable','on');    Parametric_Flag = 0;
    set(Input4_S,'visible','on','enable','on');       set(Text4_S,'visible','on');
% ====================================== PERFORM SINGLE DESIGN ALGORITHM
elseif strcmp(action,'Execute_Single')==1
    tic;         % start stopwatch
    set(Run_Single,'enable','off');                   set(Input1,'enable','off');
    set(Input1_S,'enable','off');                     set(Input2_S,'enable','off');
    set(Input3_S,'enable','off');                     set(Hub_Flag,'enable','off');
    set(XCHD_in,'enable','off');                      set(XCD_in,'enable','off');
    set(XVA_in,'enable','off');                       set(XVT_in,'enable','off');
    set(f0oc_in,'enable','off');                      set(t0oc_in,'enable','off');
    set(skew_in,'enable','off');                      set(rake_in,'enable','off');
set(Input4_S,'enable','off');
    % --------------------------------- Close figures previously created
    if ishandle(Fig_P)~=0
        close(Fig_P);
    end
    if ishandle(Fig1_S)~=0
        close(Fig1_S);
    end
    if ishandle(Fig2_S)~=0
        close(Fig2_S);
    end
    if ishandle(Fig3_S)~=0
        close(Fig3_S);
    end
    % ----------------------------------------------- Derive Input Fields
    string1 = str2double(get(Input1_S,'string'));
    string2 = str2double(get(Input1,'string'));
    string3 = str2double(get(Input2_S,'string'));
    NBLADE = string1(1);          N = string1(2);              D = string1(3);
    THRUST = string2(1);          V = string2(2);              Dhub = string2(3).*D;
    MT = string2(4);              ITER = string2(5);           RHV = string2(6);
    NX = string2(7);              HR = string2(8);             HT = string2(9);
    CRP = string2(10);            rho = string2(11);           filenamek=get(Input4_S,'string');
        if length(char(filenamek)) > 20     % Warn if hub diameter is too small and filename is
too long
        set(Err_Single,'visible','on','enable','on','string','Use Filename <= 20 Characters');
        return;
    end
    H = string3(1);               dV = string3(2);
    AlphaI = string3(3);          NP = string3(4);
    R = D/2;                      Rhub = Dhub/2;      R_input = Rhub;          n = N/60;
        %three_D= 1;%true only if for 3-D file
        three_D= 0;
        if three_D==1
            Rhub = .95*Rhub;
        end
    ADVCO = V/(n*D);              CTDES = THRUST/(rho*V^2*pi*R^2/2);
    IHUB = get(Hub_Flag,'Value');
    Meanline = get(Input3_S(1),'Value');
    Thickness = get(Input3_S(2),'Value');
    XCHD0 = str2double(get(XCHD_in,'String'));
    XCD0 = str2double(get(XCD_in,'String'));
    XVA0 = str2double(get(XVA_in,'String'));
    XVT0 = str2double(get(XVT_in,'String'));
    f0oc0 = str2double(get(f0oc_in,'String'));
    t0oc0 = str2double(get(t0oc_in,'String'));
    skew0 = str2double(get(skew_in,'String'));
    rake0 = str2double(get(rake_in,'String'));
    if Dhub/D < .15       % Warn if hub diameter is too small
        set(Err_Single,'visible','on','enable','on','string','Dhub/D >= 15%');
        return;
    end
    % ----------------------------------------------- Make OpenPVL_v1_Input.txt
    XR1 = Rhub/R:(1-Rhub/R)/(NX-3):1;
    half1 = (XR1(1)+XR1(2))/2;
    half2 = (XR1(NX-3)+XR1(NX-2))/2;
    XR = [XR1(1) half1 XR1(2:NX-3) half2 XR1(NX-2)];
```

```matlab
    XCHD = pchip(XR0,XCHD0,XR);
    XCD = pchip(XR0,XCD0,XR);
    XVA = pchip(XR0,XVA0,XR);
    XVT = pchip(XR0,XVT0,XR);
    Flag1 = datestr(now,31);
    prop_name = char(filenamek);% filename
    fid = fopen([prop_name,'_Input.txt'],'w');
    fprintf(fid,'\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n',char(filenamek));
    fprintf(fid,'_Input.txt for Open_PVL_v1\n',Flag1);
    fprintf(fid,'\n%.0f \t\tNumber of Vortex Panels over the Radius\n',MT);
    fprintf(fid,'%.0f \t\tMax. Iterations in Wake Alignment\n',ITER);
    fprintf(fid,'%.0f \t\tHub Image Flag: 1=YES, 0=NO\n',IHUB);
    fprintf(fid,'%.1f \tHub Vortex Radius/Hub Radius\n',RHV);
    fprintf(fid,'%.0f \t\tNumber of Input Radii\n',NX);
    fprintf(fid,'%.0f \t\tNumber of Blades\n',NBLADE);
    fprintf(fid,'%.3f \tAdvance Coef., J, Based on Ship Speed\n',ADVCO);
    fprintf(fid,'%.3f \tDesired Thrust Coef., Ct\n',CTDES);
    fprintf(fid,'%.0f \t\tHub Unloading Factor: 0=optimum\n',HR);
    fprintf(fid,'%.0f \t\tTip Unloading Factor: 1=Reduced Loading\n',HT);
    fprintf(fid,'%.0f \t\tSwirl Cancellation Factor: 1=No Cancellation\n',CRP);
    fprintf(fid,'r/R \t  c/D \t    Cd\t    Va/Vs  Vt/Vs\n');
    for i = 1:NX
        fprintf(fid,'%6.5f  %6.5f  %6.5f  %6.2f  %6.4f\n',XR(i),XCHD(i),XCD(i),XVA(i),XVT(i));
    end
    fprintf(fid,'\nr/R is the ratio of the radial location, r, to the total length of the blade
radius, R.\n');
    fprintf(fid,'c/D is the ratio of chordlength to diameter.\n');
    fprintf(fid,'Cd is the drag coefficient.\n');
    fprintf(fid,'Va/Vs is the axial inflow velocity ratio (is 1, assuming uniform inflow in open
water).\n');
    fprintf(fid,'Vt/Vs is the tangential inflow velocity ratio (is 0, assuming uniform inflow in
open water).\n');
    fclose(fid);
    % ================================================ Call Main Function
    [CT,CP,KT,KQ,WAKE,EFFY,RC,G,VAC,VTC,UASTAR,UTSTAR,TANBC,TANBIC,CDC,CD,KTRY,KRC]...
        =Main(MT,ITER,IHUB,RHV,NX,NBLADE,ADVCO,CTDES,HR,HT,CRP,XR,XCHD,XCD,XVA,XVT,filenamek);
    % ----------------------------------------- Create Graphical Reports
    Fig1_S = figure('units','normalized','position',[.01 .06 .4 .3],'name',...
        'Graphical Report','numbertitle','off');
    subplot(2,2,1);
    plot(RC,G);
    xlabel('r/R');          ylabel('Non-Dimensional Circulation');      grid on;
    TitleString=strcat('J=',num2str(ADVCO,'%10.3f'),'; Ct=',num2str(CT(KTRY),'%10.3f'),...
        '; Kt=',num2str(KT(KTRY),'%10.3f'),'; Kq=',num2str(KQ(KTRY),'%10.3f'),...
        '; \eta=',num2str(EFFY(KTRY),'%10.3f'));
    title(TitleString);
    subplot(2,2,2);
    plot(RC,VAC,'-b',RC,VTC,'--b',RC,UASTAR,'-.r',RC,UTSTAR,':r');
    xlabel('r/R');          legend('Va/Vs','Vt/Vs','Ua*/Vs','Ut*/Vs');  grid on;
    subplot(2,2,3);
    plot(RC,TANBC,'--b',RC,TANBIC,'-r');
    xlabel('r/R');          ylabel('Degrees');      grid on;        legend('beta','betaI');
    subplot(2,2,4);
    plot(RC,CDC);
    xlabel('r/R');          ylabel('c/D');          grid on;
    % --------------------------------- Propeller Performance Calculation
    w = 2*pi*n;         % Angular velocity w
    for k = 1:MT
        Vstar(k) = sqrt((VAC(k)+UASTAR(k))^2 + (w*R*RC(k)+VTC(k)+UTSTAR(k))^2);
        Gamma(k) = G(k)*2*pi*R*V;
        Cl(k) = 2*Gamma(k) / (Vstar(k)*CDC(k)*D);
        dBetai(k) = atand((tand(TANBIC(k))*w*RC(k)*R+dV)/(w*RC(k)*R))...
                    -atand((tand(TANBIC(k))*w*RC(k)*R-dV)/(w*RC(k)*R));
        Sigma(k) = (101000+rho*9.81*(H-RC(k)*R)-2500)/(rho*Vstar(k)^2/2); %Cavitation Number
    end
    KCl=[Cl Cl(k)];
    f0oc = pchip(XR0,f0oc0,RC).*Cl;  %F0oc = pchip(XR0,f0oc0,KRC).*KCl   % Scale camber ratio
with lift coefficient
    t0oc = pchip(XR0,t0oc0,RC);      %T0oc = pchip(XR0,t0oc0,KRC);
```

69

```matlab
        F0oc = [f0oc f0oc(MT)];        T0oc = [t0oc t0oc(MT)];
        prop_name = char(filenamek);% filename
        fid = fopen([prop_name,'_Performance.txt'],'w');
        fprintf(fid,'\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n',prop_name);
        fprintf(fid,'_Performance.txt for Open_PVL_v1\n');
        fprintf(fid,'\t\t\t\tPropeller Performance Table\n');
        fprintf(fid,' r/R\t\tV*\t beta\t betai\t  Gamma\t\tCl\t sigma\td_alpha\n');
        for k = 1:MT
            fprintf(fid,'%.3f\t %.3f\t %.2f\t %.2f\t %.4f\t %.3f\t %.3f\t %.2f\n'...
            ,RC(k),Vstar(k),TANBC(k),TANBIC(k),Gamma(k),Cl(k),Sigma(k),dBetai(k));
        end
        fprintf(fid,'\nr/R is the ratio of the radial location, r, to the total length of the blade
radius, R.\n');
        fprintf(fid,'V* is the total inflow velocity.\n');
        fprintf(fid,'beta is the undisturbed flow angle.\n');
        fprintf(fid,'betai is the hydrodynamic Pitch angle.\n');
        fprintf(fid,'Gamma is the vortex sheet strength.\n');
        fprintf(fid,'Cl is the lift coefficient.\n');
        fprintf(fid,'Sigma is the cavitation number.\n');
        fprintf(fid,'d_alpha is the inflow variation bucket width.\n');
        fclose(fid);
        % ----------------------------------------------- Geometry Calculation
        skew = pchip(XR0,skew0,RC);   Skew = pchip(XR0,skew0,KRC);
        rake = pchip(XR0,rake0,RC);   Rake = pchip(XR0,rake0,KRC);
        fid = fopen([prop_name,'_Geometry.dat'],'w');
        fprintf(fid,'\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n',char(filenamek));
        fprintf(fid,'_Geometry.dat for Open_PVL_v1\n');
        fprintf(fid,'Propeller Diameter = %.1f m\n',D);
        fprintf(fid,'Number of Blades = %.0f\n',NBLADE);
        fprintf(fid,'Propeller Speed= %.0f RPM\n',N);
        fprintf(fid,'Propeller Hub Diameter = %.2f m\n',Dhub);
        if Meanline==1
            fprintf(fid,'Meanline Type: NACA a=0.8\n');
        elseif Meanline==2
            fprintf(fid,'Meanline Type: Parabolic\n');
        end
        if Thickness==1
            fprintf(fid,'Thickness Type: NACA 65A010\n\n');
        elseif Thickness==2
            fprintf(fid,'Thickness Type: Elliptical\n\n');
        elseif Thickness==3
            fprintf(fid,'Thickness Type: Parabolic\n\n');
        elseif Thickness==4
            fprintf(fid,'Thickness Type: NACA Modified for 3-D Printing\n\n');
        end
        fprintf(fid,' r/R\t P/D\t  c/D\t  fo/c\t  to/c\t rake\t skew\n');
        for i = 1:MT
            ThetaP(i) = TANBIC(i) + AlphaI;                   % Pitch angle
            PitchOD(i) = tand(ThetaP(i))*pi*RC(i);            % P/D
            fprintf(fid, '%.3f\t %.2f\t %.3f\t %.4f\t %.4f\t %.1f\t %.3f\n'...
            ,RC(i),PitchOD(i),CDC(i),f0oc(i),t0oc(i),rake(i),skew(i));
        end
        fprintf(fid,'\nr/R is the ratio of the radial location, r, to the total length of the blade
radius, R.\n');
        fprintf(fid,'P/D is the ratio of pitch over diameter.\n');
        fprintf(fid,'c/D is the ratio of chordlength to diameter.\n');
        fprintf(fid,'fo/C is the ratio of camber to chordlength.\n');
        fprintf(fid,'to/C is the ratio of thickness to chordlength.\n');
        fclose(fid);
        % ----------------------------------------------------- BASIC SHAPE
        c = CDC.*D;
        r = RC.*R;
        theta = 0:360/NBLADE:360;   % Angles between blades
        for i = 1:MT
            for j = 1:NP
                x1(i,j) = c(i)/2-c(i)/(NP-1)*(j-1);
                station(1,j) = 1/(NP-1)*(j-1);              % For NACA foil
                z1(i,j,1) = sqrt(r(i)^2-x1(i,j)^2);
            end
```

```matlab
    end
% ------------------------------------------------- MEANLINE & THICKNESS
    x = [0 .5 .75 1.25 2.5 5 7.5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 ...
        80 85 90 95 100]./100;
    if Meanline==1          % NACA a=0.8 meanline is chosen
        foc = [0 .287 .404 .616 1.077 1.841 2.483 3.043 3.985 4.748 5.367 5.863 6.248...
            6.528 6.709 6.79 6.77 6.644 6.405 6.037 5.514 4.771 3.683 2.435 1.163 0]./100;
        fscale = f0oc./max(foc);        % Scale for camber
        dfdx0 = [.48535 .44925 .40359 .34104 .27718 .23868 .21050 .16892...
            .13734 .11101 .08775 .06634 .04601 .02613 .00620 -.01433 -.03611...
            -.06010 -.08790 -.12311 -.18412 -.23921 -.25583 -.24904 -.20385];
        for i = 1:MT
            for j = 1:NP
                f(i,:) = pchip(x,foc.*fscale(i).*c(i),station);
                dfdx(i,:) = pchip(x(2:end),dfdx0,station);
            end
        end
    elseif Meanline==2      % Parabolic meanline is chosen
        for i = 1:MT
            for j = 1:NP
                f(i,j) = f0oc(i)*c(i)*(1-(2*x1(i,j)/c(i))^2);
                dfdx(i,j) = -8*f0oc(i)*x1(i,j)/c(i);
            end
        end
    end
    if Thickness==1         % NACA 65A010 thickness form is chosen
        toc_65 = [0 .765 .928 1.183 1.623 2.182 2.65 3.04 3.658 4.127 4.483 4.742 4.912...
            4.995 4.983 4.863 4.632 4.304 3.899 3.432 2.912 2.352 1.771 1.188 .604 .021]./100;
        tscale = t0oc./2./max(toc_65);      % Scale for thickness
        for i = 1:MT
            for j = 1:NP
                t(i,:) = pchip(x,toc_65.*tscale(i).*c(i),station);
            end
        end
    elseif Thickness==2     % Elliptical thickness form is chosen
        for i = 1:MT
            for j = 1:NP
                t(i,j) = t0oc(i)*c(i)*real(sqrt(1-(2*x1(i,j)/c(i))^2));
            end
        end
    elseif Thickness==3     % Parabolic thickness form is chosen
        for i = 1:MT
            for j = 1:NP
                t(i,j) = t0oc(i)*c(i)*(1-(2*x1(i,j)/c(i))^2);
            end
        end
    elseif Thickness==4         % NACA 65A010 thickness form is modified for 3-D Printing
        tfactor=3;%thickness factor makes small model blades thick enough for printing, also the
blade tip is artificially thickened
        toc_3_D = tfactor.*[0 .765 .928 1.183 1.623 2.182 2.65 3.04 3.658 4.127 4.483 4.742
4.912...
            4.995 4.983 4.863 4.632 4.304 3.899 3.432 2.912 2.352 1.771 1.188 .604 .21]./100;
        tscale = t0oc./2./max(toc_3_D);     % Scale for thickness
        for i = 1:MT
            for j = 1:NP
                t(i,:) = pchip(x,toc_3_D.*tscale(i).*c(i),station);
            end
        end
    end
    % ------------------------------------------------- CAMBER & THICKNESS
    for i = 1:MT
        for j = 1:NP
            xu(i,j) = x1(i,j)+t(i,j)*sin(atan(dfdx(i,j)));
            xl(i,j) = x1(i,j)-t(i,j)*sin(atan(dfdx(i,j)));
            yu(i,j) = f(i,j)+t(i,j)*cos(atan(dfdx(i,j)));
            yl(i,j) = f(i,j)-t(i,j)*cos(atan(dfdx(i,j)));
            if (isreal(yu(i,j))==0)||(isreal(yl(i,j))==0)
                if Single_Flag==1
                    set(Err_Single,'visible','on','enable','on','string','Error in OpenPVL_v1.');
```

```matlab
                    if ishandle(Fig1_S)~=0
                        close(Fig1_S);
                    end
                    return;
                end
            end
        end
end
% ------------------------------------------------ PITCH, SKEW & RAKE
yrake = rake.*D;  % Rake: translation in y direction
for i = 1:MT
    for j = 1:NP
        xup(i,j,1)=xu(i,j)*cosd(ThetaP(i))-yu(i,j)*sind(ThetaP(i));
        xlp(i,j,1)=xl(i,j)*cosd(ThetaP(i))-yl(i,j)*sind(ThetaP(i));
        yup(i,j,1)=xu(i,j)*sind(ThetaP(i))+yu(i,j)*cosd(ThetaP(i));
        ylp(i,j,1)=xl(i,j)*sind(ThetaP(i))+yl(i,j)*cosd(ThetaP(i));
        zup(i,j,1) = sqrt(r(i)^2-xup(i,j,1)^2);
        zlp(i,j,1) = sqrt(r(i)^2-xlp(i,j,1)^2);

        xus(i,j,1)=xup(i,j)*cosd(skew(i))-zup(i,j)*sind(skew(i));
        xls(i,j,1)=xlp(i,j)*cosd(skew(i))-zlp(i,j)*sind(skew(i));
        yus(i,j,1)=yup(i,j);
        yls(i,j,1)=ylp(i,j);
        zus(i,j,1)=xup(i,j)*sind(skew(i))+zup(i,j)*cosd(skew(i));
        zls(i,j,1)=xlp(i,j)*sind(skew(i))+zlp(i,j)*cosd(skew(i));

        xur(i,j,1)=xus(i,j);
        xlr(i,j,1)=xls(i,j);
        yur(i,j,1)=yus(i,j)-yrake(i);
        ylr(i,j,1)=yls(i,j)-yrake(i);
        zur(i,j,1)=zus(i,j,1);
        zlr(i,j,1)=zls(i,j,1);

        for k = 2:length(theta)-1
            xur(i,j,k) = xur(i,j,1)*cosd(theta(k))-zur(i,j,1)*sind(theta(k));
            xlr(i,j,k) = xlr(i,j,1)*cosd(theta(k))-zlr(i,j,1)*sind(theta(k));
            yur(i,j,k) = yur(i,j,1);
            ylr(i,j,k) = ylr(i,j,1);
            zur(i,j,k) = xur(i,j,1)*sind(theta(k))+zur(i,j,1)*cosd(theta(k));
            zlr(i,j,k) = xlr(i,j,1)*sind(theta(k))+zlr(i,j,1)*cosd(theta(k));
        end
    end
end
% ------------------------- Create Figure for 2D Propeller Blade Image
Fig2_S = figure('units','normalized','position',[0.31 .06 .4 .3],'name',...
    'Blade Image','numbertitle','off');
style=['r' 'g' 'b' 'm' 'k'];
str_prefix = {'r/R = '};
flag=1;
for i = 1:ceil(MT/5):MT
    plot(xur(i,:,1),yur(i,:,1),style(flag));
    str_legend(flag)=strcat(str_prefix,num2str(RC(i)));
    hold on;
    flag = flag+1;
end
flag=1;
for i = 1:ceil(MT/5):MT
    plot(xlr(i,:,1),ylr(i,:,1),style(flag));
    hold on;
    flag = flag+1;
end
legend(str_legend,'location','northwest');
axis equal;    grid on;    xlabel('X (m)');    ylabel('Y (m)');    hold off;
% ------------------------------------------- Create 3D Propeller Image
Fig3_S = figure('units','normalized','position',[.61 .06 .4 .3],...
    'name','Propeller Image','numbertitle','off');
for k = 1:NBLADE
    surf(xur(:,:,k),yur(:,:,k),zur(:,:,k));        hold on;
    surf(xlr(:,:,k),ylr(:,:,k),zlr(:,:,k));        hold on;
```

72

```matlab
    end
    tick = 0:15:90;
    [xh0,yh0,zh0] = cylinder(R_input*sind(tick),50);
    surf(yh0,zh0.*.3+min(ylp(1,:,1))-.3,xh0);
    [xh1,yh1,zh1] = cylinder(R_input,50);
    surf(yh1, zh1+min(ylp(1,:,1)), xh1);
    hold off;        colormap gray;          grid on;         axis equal;
    xlabel('X');    ylabel('Y');            zlabel('Z');
    set(New_Single,'visible','on','enable','on');
    figure(Fig_Main);
%START OF RHINO™ SCRIPTING
%turn on 3-D mode search 'three_D'
prop_name = char(filenamek);% filename
numSections = length(RC);
%Blade geometry in X, Y, Z coordinates
    LocalChord = [xur(:,:,1) fliplr(xlr(:,:,1))];
    LocalHeight = [yur(:,:,1) fliplr(ylr(:,:,1))];
    LocalSpan = [zur(:,:,1) fliplr(zlr(:,:,1))];
%%%% Output text files of each section that can be read in by a CAD program
m2in=100/2.54; %convert from meters to inches
%m2in=1;%turns off conversion
Diam = D*m2in;%          Diameter of propeller (from m to in)
hubRad = Diam/2*RC(1);
aPitch=atan(PitchOD(1)/(pi*RC(1)));
hub_ht = CDC(1)/2*Diam*sin(aPitch);%     Half of propeller hub height
R_t_edge = t(1)*20;%Radius of fillet curve, currently based on thickness at hub
hub_h = hub_ht*2;%changed to accommodate fillet size
%CREATE RHINO™ SCRIPT (Modified Jordan Code, used with permission)
fid = fopen([prop_name,'_CADfull.txt'],'w');
    fprintf(fid,'! ');
    for section=1 %changed 26JUN
    fprintf(fid,'Curve \n');
    for j=1:length(LocalChord(section,:))
        fprintf(fid,'%f,%f,%f\n',m2in*LocalChord(section,j),...
            .95*m2in*LocalSpan(section,j),m2in*LocalHeight(section,j) );
    end
end
for section=2:numSections %changed 26JUN
    fprintf(fid,'Curve \n');
    for j=1:length(LocalChord(section,:))
        fprintf(fid,'%f,%f,%f\n',m2in*LocalChord(section,j),...
            m2in*LocalSpan(section,j),m2in*LocalHeight(section,j) );
    end
end
fprintf(fid,'SelNone\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'-Patch\n');
fprintf(fid,'USpans=10\n');
fprintf(fid,'VSpans=10\n');
fprintf(fid,'Stiffness=0.0001\n');
fprintf(fid,'AutomaticTrim=Yes\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNone\n');
fprintf(fid,'SelCrv\n');
fprintf(fid,'-Loft Tight RebuildCount=10\n');
fprintf(fid,'enter\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNone\n');
fprintf(fid,'SelCrv\n');
fprintf(fid,'Delete\n');
fprintf(fid,'SelAll\n');
fprintf(fid,'Join\n');
fprintf(fid,'Dir\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'Flip\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'Rotate\n');
fprintf(fid,'Copy=Yes\n');
```

```matlab
fprintf(fid,'0,0,0\n');
for blade=2:NBLADE
fprintf(fid,'%f\n',(blade-1)*(360/NBLADE));
end

fprintf(fid,'enter\n');
fprintf(fid,'SelNone\n');
fprintf(fid,'Circle\n');
fprintf(fid,'0,0,0\n');
fprintf(fid,'%f\n',hubRad);
fprintf(fid,'SelNone\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'ExtrudeCrv BothSides=Yes Cap=Yes DeleteInput=Yes\n ');
fprintf(fid,'%f\n',1.25*hub_h);
fprintf(fid,'SelNone\n');
% fprintf(fid,'Paraboloid MarkFocus=Yes Cap=Yes\n');%new
% fprintf(fid,'0,0,1\n');
% fprintf(fid,'0,0,.75\n');
% fprintf(fid,'0,.92,.82\n');
% fprintf(fid,'enter\n');
 fprintf(fid,'BooleanUnion\n');
 fprintf(fid,'SelAll\n');
 fprintf(fid,'enter\n');

fprintf(fid,'-Mesh\n');
fprintf(fid,'SelAll\n');
fprintf(fid,'enter\n');
fprintf(fid,'enter\n');

fprintf(fid,'Weld\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');
fprintf(fid,'180\n');
fprintf(fid,'enter\n');
fprintf(fid,'UnifyMeshNormals\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNakedMeshEdgePt\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');

fprintf(fid,'Zoom All Extents\n');
fprintf(fid,'enter\n');
fclose(fid);

fid = fopen([prop_name '_CADblade.txt'],'w');
    fprintf(fid,'! ');
for section=1:numSections
    fprintf(fid,'Curve \n');
    for j=1:length(LocalChord(section,:))
        fprintf(fid,'%f,%f,%f\n',m2in*LocalChord(section,j),...
            m2in*LocalSpan(section,j),m2in*LocalHeight(section,j) );
    end
end
fprintf(fid,'SelNone\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'-Patch\n');
fprintf(fid,'USpans=10\n');
fprintf(fid,'VSpans=10\n');
fprintf(fid,'Stiffness=0.0001\n');
fprintf(fid,'AutomaticTrim=Yes\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNone\n');
fprintf(fid,'SelCrv\n');
fprintf(fid,'-Loft Tight RebuildCount=10\n');
fprintf(fid,'enter\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNone\n');
fprintf(fid,'SelCrv\n');
```

```matlab
fprintf(fid,'Delete\n');
fprintf(fid,'SelAll\n');
fprintf(fid,'Join\n');
%new code follows:
fprintf(fid,'Dir\n');
fprintf(fid,'enter\n');
fprintf(fid,'Flip\n');
fprintf(fid,'enter\n');

fprintf(fid,'SelNone\n');
fprintf(fid,'Circle\n');
fprintf(fid,'0,0,0\n');
fprintf(fid,'%f\n',hubRad);
fprintf(fid,'SelNone\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'ExtrudeCrv BothSides=Yes Cap=Yes DeleteInput=Yes\n ');
fprintf(fid,'%f\n',hub_h);
fprintf(fid,'SelNone\n');
fprintf(fid,'BooleanUnion\n');
fprintf(fid,'SelAll\n');
fprintf(fid,'enter\n');

fprintf(fid,'-Mesh\n');
fprintf(fid,'SelAll\n');
fprintf(fid,'enter\n');
fprintf(fid,'enter\n');

fprintf(fid,'Weld\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');
fprintf(fid,'180\n');
fprintf(fid,'enter\n');
fprintf(fid,'UnifyMeshNormals\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');
fprintf(fid,'SelNakedMeshEdgePt\n');
fprintf(fid,'SelLast\n');
fprintf(fid,'enter\n');

fprintf(fid,'Zoom All Extents\n');
fprintf(fid,'enter\n');
fclose(fid);
toc;
disp([prop_name ' RHINO™ script files created.'])
disp('Please open RHINO™3d with a new (inches) document.')
disp('Use ReadCommandFile to generate the propeller.')
end     % =============================================== Do not delete
% =============================================== Main Function Code
function [CT,CP,KT,KQ,WAKE,EFFY,RC,G,VAC,VTC,UASTAR,UTSTAR,TANBC,TANBIC,CDC,CD,KTRY,KRC]=...
    Main(MT,ITER,IHUB,RHV,NX,NBLADE,ADVCO,CTDES,HR,HT,CRP,XR,XCHD,XCD,XVA,XVT,filenamek)
    global Parametric_Flag Single_Flag
    % ------------------------------------------- FORTRAN Function VOLWK
    YW = XR.*XVA;
    YDX = trapz(XR,YW);
    WAKE = 2*YDX/(1-XR(1)^2);
    % ------------------------------------------------------- End of VOLWK
    XRC = 1-sqrt(1-XR);
    DEL = pi/(2*MT);              % Compute cosine spaced vortex radii
    HRR = 0.5*(XR(NX)-XR(1));
    for i=1:MT+1
        RV(i) = XR(1)+HRR*(1-cos(2*(i-1)*DEL));
    end
    VAV = pchip(XR,XVA,RV);
    VTV = pchip(XR,XVT,RV);
    TANBV = VAV./((pi.*RV./ADVCO)+VTV);
    VBAV = VTV.*TANBV./VAV;
    % Cosine spaced control point radii: Evaluate c/D,Va,Vt,tanB,Cd,Vt*,tanB/Va
    for i=1:MT
        RC(i) = XR(1)+HRR*(1-cos((2*i-1)*DEL));
```

75

```matlab
        RCWG(i) = 1-sqrt(1-RC(i));
    end
    KRC=[RC 1];
    CDC = pchip(XRC,XCHD,RCWG);
    CD = pchip(XR,XCD,RC);
    VAC = pchip(XR,XVA,RC);
    VTC = pchip(XR,XVT,RC);
    TANBC = VAC./(pi.*RC./ADVCO+VTC);
    VBAC = VTC.*TANBC./VAC;
    % First estimation of tanBi based on 90% of actuator disk efficiecy
    EDISK = 1.8/(1+sqrt(1+CTDES/WAKE^2));
    TANBXV = TANBV.*sqrt(WAKE./(VAV-VBAV))/EDISK;
    TANBXC = TANBC.*sqrt(WAKE./(VAC-VBAC))/EDISK;
    % Unload hub and tip as specified by HR and HT
    RM = (XR(1)+XR(NX))/2;
    for i=1:MT+1
        if RV(i)<RM
            HRF=HR;
        else
            HRF=HT;
        end
        DTANB = HRF*(TANBXV(i)-TANBV(i))*((RV(i)-RM)/(XR(1)-RM))^2;
        TANBXV(i) = TANBXV(i)-DTANB;
    end
    for i=1:MT
        if RC(i)<RM
            HRF=HR;
        else
            HRF=HT;
        end
        DTANB = HRF*(TANBXC(i)-TANBC(i))*((RC(i)-RM)/(XR(1)-RM))^2;
        TANBXC(i) = TANBXC(i)-DTANB;
    end

    % Iterations to scale tanBi to get desired value of thrust coefficient
    for KTRY=1:ITER
        if KTRY==1
            T(KTRY) = 1;
        elseif KTRY==2
            T(KTRY) = 1+(CTDES-CT(1))/(5*CTDES);
        elseif KTRY>2
            if CT(KTRY-1)-CT(KTRY-2)==0
                break
            else
                T(KTRY)=T(KTRY-1)+(T(KTRY-1)-T(KTRY-2))*(CTDES-CT(KTRY-1))/...
                    (CT(KTRY-1)-CT(KTRY-2));
            end
        end
        TANBIV = T(KTRY).*TANBXV;
        TANBIC = T(KTRY).*TANBXC;
        % Compute axial and tangential horseshoe influence coefficients
        for i=1:MT
            RCW = RC(i);
            for j = 1:MT+1
                RVW = RV(j);
                TANBIW = TANBIV(j);     % induction of trailing vortices shed at RV(N)
                [UAIF,UTIF]=WRENCH(NBLADE,TANBIW,RCW,RVW);
                if (isnan(UAIF)==1)||(isnan(UTIF)==1)
                    if Single_Flag==1
                        set(Err_Single,'visible','on','enable','on','string','Error in
OpenPVL_v1.');
                    end
                    return;
                end
                UAW(j) = -UAIF/(2*(RC(i)-RV(j)));
                UTIF = UTIF*CRP;
                UTW(j) = UTIF/(2*(RC(i)-RV(j)));
                % Induction of corresponding hub-image trailing vorticies
                if IHUB==1
```

```matlab
                RVW = XR(1)^2/RV(j);
                TANBIW = TANBIV(1)*RV(1)/RVW;
                [UAIF,UTIF]=WRENCH(NBLADE,TANBIW,RCW,RVW);
                if (isnan(UAIF)==1)||(isnan(UTIF)==1)
                    if Single_Flag==1
                        set(Err_Single,'visible','on','enable','on',...
                            'string','Error in OpenPVL_v1.');
                    end
                    return;
                end
                UAW(j) = UAW(j)+UAIF/(2*(RC(i)-RVW));
                UTIF = UTIF*CRP;
                UTW(j) = UTW(j)-UTIF/(2*(RC(i)-RVW));
            end
        end
        % Final step in building influence functions
        for k=1:MT
            UAHIF(i,k) = UAW(k+1)-UAW(k);
            UTHIF(i,k) = UTW(k+1)-UTW(k);
        end
    end
    % Solve simutaneous equations for circulation strengths G(i)
    for m=1:MT
        B(m) = VAC(m)*((TANBIC(m)/TANBC(m))-1);
        for n=1:MT
            A(m,n) = UAHIF(m,n)-UTHIF(m,n)*TANBIC(m);
        end
    end
    % ==================================== FORTRAN Subroutine SIMEQN
    NEQ = length(B);
    IERR = 1;
    % Find |maximum| element in each row and exit if a zero row is detected
    for i=1:NEQ
        IPIVOT(i) = i;
        ROWMAX = 0;
        for j=1:NEQ
            ROWMAX = max(ROWMAX, abs(A(i,j)));
        end
        if ROWMAX==0
            fprintf('Matrix is Singular-1.\n')
            G = NaN;     % Must return something to avoid error warning
            if Single_Flag==1
                set(Err_Single,'visible','on','enable','on','string','Error in OpenPVL_v1.');
            end
            return;
        end
        D(i) = ROWMAX;
    end
    NM1=NEQ-1;
    if NM1>0        % Otherwise special case of one equation
        for k=1:NM1
            j = k;
            KP1 = k+1;
            IP = IPIVOT(k);
            COLMAX = abs(A(IP,k))/D(IP);
            for i=KP1:NEQ
                IP = IPIVOT(i);
                AWIKOV = abs(A(IP,k))/D(IP);
                if AWIKOV>COLMAX
                    COLMAX = AWIKOV;
                    j=i;
                end
            end
            if COLMAX==0
                fprintf('Matrix is Singular-2.\n')
                G = NaN;   % Must return something to avoid error warning
                if Single_Flag==1
                    set(Err_Single,'visible','on','enable','on','string','Error in
OpenPVL_v1.');
```

```matlab
            end
            return;
        end
        IPK = IPIVOT(j);
        IPIVOT(j) = IPIVOT(k);
        IPIVOT(k) = IPK;
        for i=KP1:NEQ
            IP = IPIVOT(i);
            A(IP,k) = A(IP,k)/A(IPK,k);
            RATIO = -A(IP,k);
            for j=KP1:NEQ
                A(IP,j) = RATIO*A(IPK,j)+A(IP,j);
            end
        end
    end
    if A(IP,NEQ)==0
        fprintf('Matrix is Singular-3.\n')
        G = NaN;  % Must return something to avoid error warning
        if Single_Flag==1
            set(Err_Single,'visible','on','enable','on','string','Error in OpenPVL_v1.');
        end
        return;
    end
end
IERR = 0;        % Matrix survived singular test
if NEQ==1        % Back substitute to obtain solution (G)
    G(1) = B(1)/A(1,1);
else
    IP = IPIVOT(1);
    G(1) = B(IP);
    for k=2:NEQ
        IP = IPIVOT(k);
        KM1 = k-1;
        SUMM = 0;
        for j=1:KM1
            SUMM = A(IP,j)*G(j)+SUMM;
        end
        G(k) = B(IP)-SUMM;
    end
    G(NEQ) = G(NEQ)/A(IP,NEQ);
    k = NEQ;
    for NP1MK=2:NEQ
        KP1 = k;
        k = k-1;
        IP = IPIVOT(k);
        SUMM = 0;
        for j=KP1:NEQ
            SUMM = A(IP,j)*G(j)+SUMM;
        end
        G(k) = (G(k)-SUMM)/A(IP,k);
    end
end
% =================================================== End of SIMEQN
if IERR==1  % Matrix is singular
    if Single_Flag==1
        set(Err_Single,'visible','on','enable','on','string','Error in OpenPVL_v1.');
    end
    return;
end
% Evalutate the induced velocities from the circulation G(i)
for p=1:MT
    UASTAR(p) = 0;
    UTSTAR(p) = 0;
    for q=1:MT
        UASTAR(p) = UASTAR(p)+G(q)*UAHIF(p,q);
        UTSTAR(p) = UTSTAR(p)+G(q)*UTHIF(p,q);
    end
end
% ===================================== FORTRAN Subroutine FORCES
```

```matlab
            LD = 0;      % Default: Input is Cd, not L/D
            if CD>1
                LD = 1;  % CD> 1 means the input is L/D
            end
            CT(KTRY) = 0;
            CQ(KTRY) = 0;
            for m=1:MT
                DR = RV(m+1)-RV(m);
                VTSTAR = VAC(m)/TANBC(m)+UTSTAR(m);
                VASTAR = VAC(m)+UASTAR(m);
                VSTAR = sqrt(VTSTAR^2+VASTAR^2);
                if LD==0
                    DVISC = (VSTAR^2*CDC(m)*CD(m))/(2*pi);
                else
                    FKJ = VSTAR*G(m);
                    DVISC = FKJ/CD(m);
                end
                CT(KTRY) = CT(KTRY)+(VTSTAR*G(m)-DVISC*VASTAR/VSTAR)*DR;
                CQ(KTRY) = CQ(KTRY)+(VASTAR*G(m)+DVISC*VTSTAR/VSTAR)*RC(m)*DR;
            end
            if IHUB~=0
                CTH = .5*(log(1/RHV)+3)*(NBLADE*G(1))^2;
            else
                CTH = 0;
            end
            CT(KTRY) = CT(KTRY)*4*NBLADE-CTH;
            CQ(KTRY) = CQ(KTRY)*2*NBLADE;
            CP(KTRY) = CQ(KTRY)*2*pi/ADVCO; %power coefficient
            KT(KTRY) = CT(KTRY)*ADVCO^2*pi/8;
            KQ(KTRY) = CQ(KTRY)*ADVCO^2*pi/8;
            EFFY(KTRY) = CT(KTRY)*WAKE/CP(KTRY);
            for i=1:length(RC)
                if (isreal(TANBIC(i))==0)||(isreal(EFFY(KTRY))==0)||(EFFY(KTRY)<=0)
                    TANBIC(i) = NaN;
                    EFFY(KTRY) = NaN;
                end
            end
            % ================================================= End of FORCES
            if abs(CT(KTRY)-CTDES)<(5e-6)
                break
            end
        end             % For KTRY=1:ITER.  Do not delete
    if IERR==1      % Stop run if matrix is singular
        fprintf('Matrix is Singular. Run Terminated.\n');
        return;
    else
        if Single_Flag==1
            TANBC = atand(TANBC);
            TANBIC = atand(TANBIC);
            prop_name = char(filenamek);
            fid = fopen([prop_name,'_Output.txt'],'w');
            fprintf(fid,'\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n',char(filenamek));
            fprintf(fid,'_Output.txt for Open_PVL_v1\n');
            fprintf(fid,'\t\t\t\tOutput Table\n');
            fprintf(fid,'Ct= % 5.4f\n' ,CT(KTRY));
            fprintf(fid,'Cp= % 5.4f\n' ,CP(KTRY));
            fprintf(fid,'Kt= % 5.4f\n' ,KT(KTRY));
            fprintf(fid,'Kq= % 5.4f\n' ,KQ(KTRY));
            fprintf(fid,'Va/Vs= % 5.4f\n' ,WAKE);
            fprintf(fid,'Efficiency= %5.4f\n' ,EFFY(KTRY));
            fprintf(fid,' r/R\t  \tG\t\t  Va\t  Vt\t  Ua\t  \tUt\t \tbeta\tbetai\t c/D\t
Cd\t\n');
            for i = 1:length(RC)
            fprintf(fid,'%5.5f  %5.6f  %5.5f  %5.4f  %5.5f  %5.5f  %5.3f  %5.3f  %5.5f
%5.5f\n',...
                RC(i),G(i),VAC(i),VTC(i),UASTAR(i),UTSTAR(i),TANBC(i),TANBIC(i),CDC(i),CD(i));
            end
            fprintf(fid,'\nCt is the required thrust.\n');
            fprintf(fid,'Cp is the power coefficient. Cp = Cq*2*pi/J.\n');
```

79

```matlab
            fprintf(fid,'Kt is the thrust coefficient. Kt = Ct*J^2*pi/8, where J is the advance
coefficient.\n');
            fprintf(fid,'Kq is the torque coefficient. Kq = Cq*J^2*pi/8.\n');
            fprintf(fid,'Va/Vs is the axial inflow velocity ratio (is 1, assuming uniform inflow
in open water).\n');
            fprintf(fid,'Efficiency is the ratio of required thrust times the wake to the power
coefficient.\n');
            fprintf(fid,'r/R is the ratio of the radial location, r, to the total length of the
blade radius, R.\n');
            fprintf(fid,'G is the non-dimensional circulation.\n');
            fprintf(fid,'Va is the axial inflow velocity.\n');
            fprintf(fid,'Vt is the tangential inflow velocity.\n');
            fprintf(fid,'Ua us the induced axial velocity.\n');
            fprintf(fid,'Ut is the induced tangential velocity.\n');
            fprintf(fid,'beta is the flow angle.\n');
            fprintf(fid,'betaI is the hydrodynamic Pitch angle.\n');
            fprintf(fid,'c/D is the ratio of chordlength to diameter.\n');
            fprintf(fid,'Cd is the drag coefficient.\n');
            fclose(fid);
        end
    end
    % =================================================== End of Main Function
% ==================================== FORTRAN Subroutine WRENCH
function [UAIF,UTIF]=WRENCH(NBLADE,TANBIW,RCW,RVW)
    if NBLADE>20     % Return infinite blade result if NBLADE>20
        if RCW>RVW
            UAIF = 0;
            UTIF = NBLADE*(RCW-RVW)/RCW;
        else
            UAIF = -NBLADE*(RCW-RVW)/(RVW*TANBIW);
            UTIF = 0;
        end
        return;
    end
    XG = 1/TANBIW;      % End of infinite blade patch
    ETA = RVW/RCW;
    H = XG/ETA;
    XS = 1+H^2;
    TW = sqrt(XS);
    V = 1+XG^2;
    W = sqrt(V);
    AE = TW-W;
    U = exp(AE);
    R = (((TW-1)/H*(XG/(W-1)))*U)^NBLADE;
    XX = (1/(2*NBLADE*XG))*((V/XS)^0.25);
    Y = ((9*XG^2)+2)/(V^1.5)+((3*H^2-2)/(XS^1.5));
    Z = 1/(24*NBLADE)*Y;
    if H>=XG
        AF = 1+1/(R-1);
        if AF==0
            UAIF = NaN;
            UTIF = NaN;
            return;
        else
            AA = XX*(1/(R-1)-Z*log(AF));
            UAIF = 2*NBLADE^2*XG*H*(1-ETA)*AA;
            UTIF = NBLADE*(1-ETA)*(1+2*NBLADE*XG*AA);
        end
    else
        if R>1e-12
            RATIO = 1/(1/R-1);
        else
            RATIO = 0;
        end
        AG = 1+RATIO;
        if AG==0
            UAIF = NaN;
            UTIF = NaN;
            return;
```

80

```
    else
        AB = -XX*(RATIO+Z*log(AG));
        UAIF = NBLADE*XG*(1-1/ETA)*(1-2*NBLADE*XG*AB);
        UTIF = 2*NBLADE^2*XG*(1-ETA)*AB;
    end
end
```

Appendix B: AUV Propeller Design Results

B.1   AUV Propeller Chordlength Coordinates

Table B-1: Chordlength distribution of an off-the-shelf propeller.

R	r/R	C	c/D
0.0200	0.0673	0.0488	0.0822
0.0300	0.1010	0.0459	0.0773
0.0400	0.1347	0.0445	0.0749
0.0500	0.1684	0.0440	0.0741
0.0600	0.2020	0.0443	0.0746
0.0700	0.2357	0.0452	0.0761
0.0800	0.2694	0.0469	0.0790
0.0900	0.3030	0.0488	0.0822
0.1000	0.3367	0.0506	0.0852
0.1100	0.3704	0.0507	0.0854
0.1200	0.4040	0.0498	0.0838
0.1300	0.4377	0.0485	0.0816
0.1400	0.4714	0.0472	0.0795
0.1500	0.5051	0.0461	0.0776
0.1600	0.5387	0.0445	0.0749
0.1700	0.5724	0.0430	0.0724
0.1800	0.6061	0.0414	0.0697
0.1900	0.6397	0.0398	0.0670
0.2000	0.6734	0.0379	0.0638
0.2100	0.7071	0.0359	0.0604
0.2200	0.7407	0.0338	0.0569
0.2300	0.7744	0.0318	0.0535
0.2400	0.8081	0.0294	0.0495
0.2500	0.8418	0.0270	0.0455
0.2600	0.8754	0.0240	0.0404
0.2700	0.9091	0.0212	0.0357
0.2800	0.9428	0.0180	0.0303
0.2900	0.9764	0.0125	0.0210
0.2950	0.9933	0.0062	0.0104
0.2970	1.0000	0.0010	0.0017

Table B-2: Chordlength distribution of an optimized AUV propeller.

r/R	c/D
0.2000	0.2269
0.2500	0.2193
0.3000	0.2148
0.4000	0.2029
0.5000	0.1885
0.6000	0.1693
0.7000	0.1545
0.8000	0.1275
0.9000	0.0801
0.9500	0.0572
1.0000	0.0032

B.2 Geometry used for the US Navy 4148 Propeller Example

```
   r/R      P/D       c/D       fo/c      to/c      rake      skew
  0.204    1.11     0.162     0.0310    0.2010    0.0       0.000
  0.236    1.11     0.176     0.0278    0.1676    0.0       0.000
  0.298    1.10     0.202     0.0211    0.1191    0.0       0.000
  0.384    1.10     0.217     0.0159    0.0939    0.0       0.000
  0.487    1.11     0.230     0.0117    0.0717    0.0       0.000
  0.600    1.11     0.231     0.0089    0.0541    0.0       0.000
  0.713    1.12     0.227     0.0071    0.0469    0.0       0.000
  0.816    1.13     0.214     0.0057    0.0405    0.0       0.000
  0.902    1.13     0.179     0.0046    0.0331    0.0       0.000
  0.964    1.14     0.121     0.0033    0.0288    0.0       0.000
  0.996    1.14     0.045     0.0004    0.0039    0.0       0.000
```
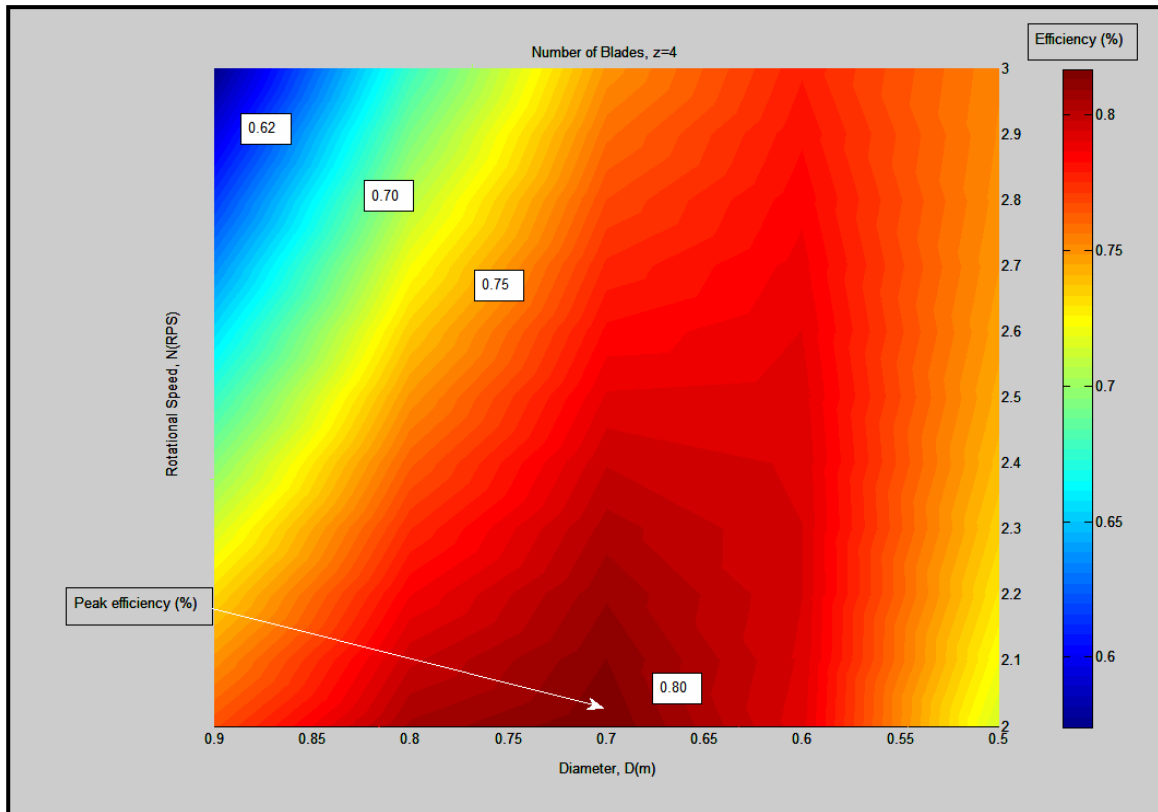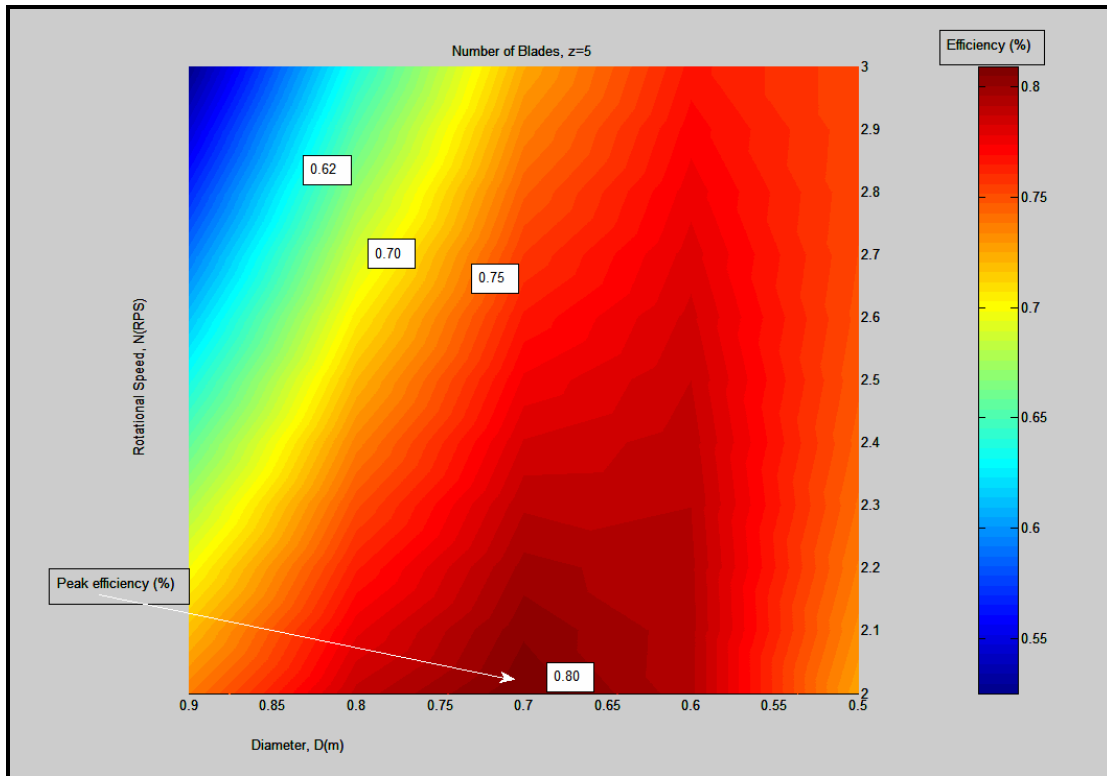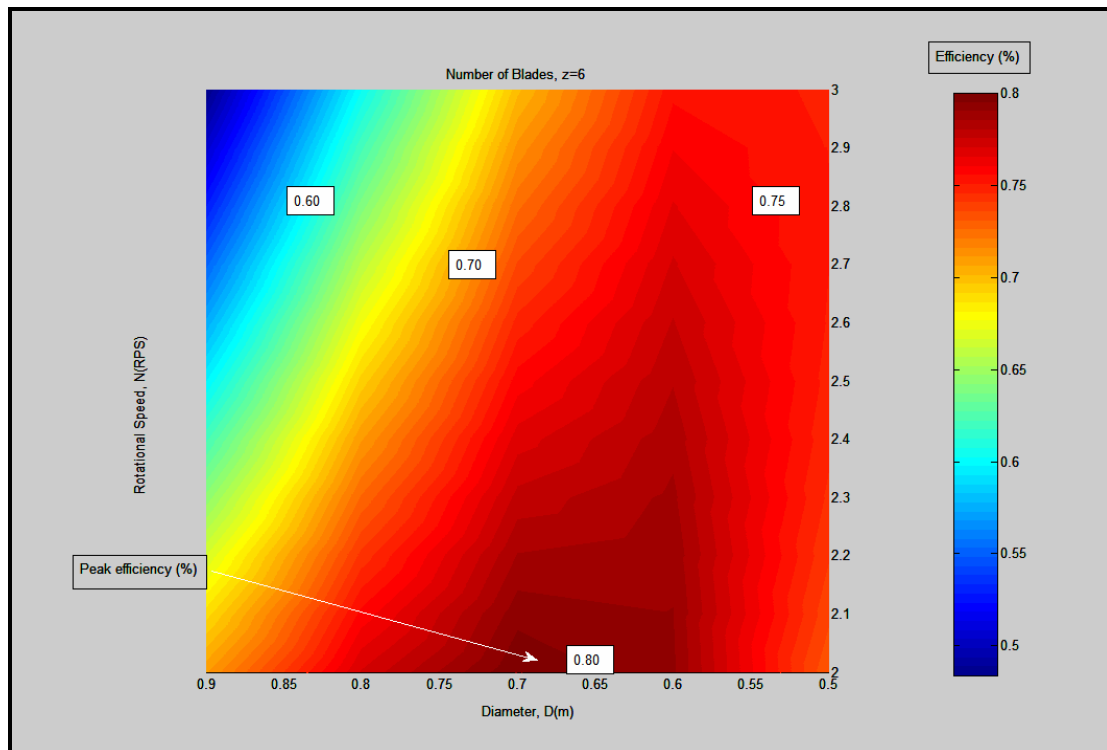
B.3 AUV basicPVL Parametric Analysis

Figure B-1: Efficiency for a 4-bladed propeller.

Figure B-2: Efficiency for a 5-bladed propeller.

Figure B-3: Efficiency for a 6-bladed propeller.

85

B.4   AUV Off-the-Shelf Propellers Tow-Tank Testing

Table B-3: Forward and reverse numerical values for the Bollard Test.

RUN	PropRPM	RPM	A	V	T	Force	Force(lbs)	Force(N)
fwdbol =								
1.0000	16.6500	5.0000	0.0152	46.6120	30.0000	1.0000	0.1987	0.8840
2.0000	33.3000	10.0000	0.0454	46.5900	29.0000	4.0000	0.7950	3.5362
3.0000	49.9500	15.0000	0.1227	46.5860	28.0400	9.0000	1.7887	7.9564
4.0000	66.6000	20.0000	0.2573	46.5540	27.0000	17.0000	3.3786	15.0287
5.0000	83.2500	25.0000	0.4859	46.5214	27.5000	27.0000	5.3660	23.8691
6.0000	99.9000	30.0000	0.8595	46.4040	28.7300	39.0000	7.7509	34.4775
7.0000	116.5500	35.0000	1.4248	46.2940	30.8400	53.0000	10.5332	46.8541
8.0000	133.2000	40.0000	2.1514	46.1220	34.7400	73.0000	14.5080	64.5349
revbol =								
1.0000	16.6500	-5.0000	0.2376	46.5710	33.5800	1.0000	0.1987	0.8840
2.0000	33.3000	-10.0000	0.3615	46.5400	33.9700	3.0000	0.5962	2.6521
3.0000	49.9500	-15.0000	0.5044	46.5120	34.2700	6.0000	1.1924	5.3042
4.0000	66.6000	-20.0000	0.7426	46.4510	34.7200	9.5000	1.8880	8.3984
5.0000	83.2500	-25.0000	1.0988	46.3700	36.1300	15.0000	2.9811	13.2606
6.0000	99.9000	-30.0000	1.4324	46.2960	42.0300	21.0000	4.1735	18.5648
7.0000	116.5500	-35.0000	2.1383	46.1520	46.6800	28.0000	5.5647	24.7531
8.0000	133.2000	-40.0000	3.0395	45.9380	47.7100	37.0000	7.3534	32.7095

Table B-4: Forward and reverse numerical values for the moving Cart Test.

RUN	PropRPM	RPM	A	V	T	Force	Cart Spd(m/s)	Force(lbs)	Force(N)

fwdcart =

RUN	PropRPM	RPM	A	V	T	Force	Cart Spd(m/s)	Force(lbs)	Force(N)
1.0000	49.9500	15.0000	0.1153	46.5720	32.0000	2.0000	0.2500	0.3975	1.7681
2.0000	49.9500	15.0000	0.0368	46.5840	33.0000	-3.0000	0.3500	0.5962	2.6521
3.0000	66.6000	20.0000	0.2329	46.5650	32.0600	8.0000	0.2500	1.5899	7.0723
4.0000	66.6000	20.0000	0.1762	46.5800	34.0000	4.0000	0.3500	0.7950	3.5362
5.0000	66.6000	20.0000	0.1000	46.5750	33.0700	-2.0000	0.4500	0.3975	1.7681
6.0000	83.2500	25.0000	0.4790	46.5220	33.0900	18.0000	0.2500	3.5773	15.9127
7.0000	83.2500	25.0000	0.4275	46.5220	35.0000	13.0000	0.3500	2.5836	11.4925
8.0000	83.2500	25.0000	0.3392	46.5500	36.0000	7.0000	0.4500	1.3912	6.1883
9.0000	83.2500	25.0000	0.2473	46.5770	36.0000	-1.0000	0.5500	0.1987	0.8840
10.0000	99.9000	30.0000	0.8849	46.4060	35.2700	33.0000	0.2500	6.5584	29.1733
11.0000	99.9000	30.0000	0.8496	46.4270	37.8300	27.0000	0.3500	5.3660	23.8691
12.0000	99.9000	30.0000	0.7325	46.4450	39.0300	21.0000	0.4500	4.1735	18.5648
13.0000	99.9000	30.0000	0.6059	46.4660	40.0000	12.5000	0.5500	2.4842	11.0505
14.0000	99.9000	30.0000	0.5023	46.5070	40.0000	2.0000	0.6500	0.3975	1.7681
15.0000	99.9000	30.0000	0.2910	46.5700	40.0000	-5.0000	0.7500	0.9937	4.4202
16.0000	116.5500	35.0000	1.5340	46.2740	38.4300	51.5000	0.2500	10.2351	45.5280
17.0000	116.5500	35.0000	1.5050	46.2690	43.3000	45.0000	0.3500	8.9433	39.7818
18.0000	116.5500	35.0000	1.4024	46.3010	44.9400	38.0000	0.4500	7.5521	33.5935
19.0000	116.5500	35.0000	1.2972	46.3070	46.0000	30.0000	0.5500	5.9622	26.5212
20.0000	116.5500	35.0000	1.0276	46.3910	46.6300	19.0000	0.6500	3.7761	16.7968
21.0000	116.5500	35.0000	0.7627	46.4450	46.7800	7.0000	0.7500	1.3912	6.1883
22.0000	116.5500	35.0000	0.5522	46.4870	46.9800	-3.0000	0.8500	0.5962	2.6521
23.0000	133.2000	40.0000	2.4441	46.0760	44.5800	70.0000	0.2500	13.9118	61.8828
24.0000	133.2000	40.0000	2.4591	46.0520	50.5900	65.0000	0.3500	12.9181	57.4626
25.0000	133.2000	40.0000	2.3672	45.9820	53.2200	58.0000	0.4500	11.5269	51.2743
26.0000	133.2000	40.0000	2.2294	46.0340	54.4000	50.0000	0.5500	9.9370	44.2020
27.0000	133.2000	40.0000	1.9677	46.1090	55.5100	40.0000	0.6500	7.9496	35.3616
28.0000	133.2000	40.0000	1.6559	46.1570	55.3500	29.0000	0.7500	5.7635	25.6371
29.0000	133.2000	40.0000	1.2676	46.2930	56.0600	15.0000	0.8500	2.9811	13.2606
30.0000	133.2000	40.0000	0.9976	46.3100	55.0000	0	0.9500	0	0
31.0000	133.2000	40.0000	0.6639	46.4000	53.7000	-5.0000	1.0500	0.9937	4.4202

revcart =

RUN	PropRPM	RPM	A	V	T	Force	Cart Spd(m/s)	Force(lbs)	Force(N)
1.0000	83.2500	25.0000	0.9600	46.3940	32.2900	11.5000	0.2500	2.2855	10.1665
2.0000	83.2500	25.0000	0.9197	46.4170	42.1800	6.5000	0.3500	1.2918	5.7463
3.0000	83.2500	25.0000	0.8764	46.4350	42.6800	-0.5000	0.4500	0.0994	0.4420
4.0000	99.9000	30.0000	1.4181	46.2980	36.9400	19.0000	0.2500	3.7761	16.7968
5.0000	99.9000	30.0000	1.5307	46.2880	43.5500	13.5000	0.3500	2.6830	11.9345
6.0000	99.9000	30.0000	1.7507	46.2190	47.3700	6.5000	0.4500	1.2918	5.7463
7.0000	99.9000	30.0000	1.2024	46.3470	47.5800	-2.5000	0.5500	0.4968	2.2101
8.0000	99.9000	30.0000	1.3644	46.3310	48.0000	3.0000	0.4900	0.5962	2.6521
9.0000	116.5500	35.0000	2.3352	46.1090	47.5100	22.5000	0.3500	4.4716	19.8909
10.0000	116.5500	35.0000	2.3075	46.1220	35.5800	37.5000	0.2500	7.4527	33.1515
11.0000	116.5500	35.0000	2.0097	46.1860	42.8800	15.0000	0.4500	2.9811	13.2606
12.0000	116.5500	35.0000	1.9280	46.2030	47.8500	6.0000	0.5500	1.1924	5.3042
13.0000	116.5500	35.0000	1.9105	46.2120	50.0200	-5.0000	0.6500	0.9937	4.4202

B.5   Actual Measurements for the 3D Printed Propeller Blade

```
Point   x     y    z_fore   z_aft
1    2.5707  1.5464  -0.0309 -0.0398
2    2.8907  0.8025  -0.5636 -0.6551
3    3    0    -1.1199 -1.2514
4    2.8907  -0.8025  -1.778  -1.8506
5    2.5707  -1.5464  -1.9488 -2.5876
6    8.2315  -1.6739  -1.5567 -1.9734
7    8.3578  -0.8412  -1.2415 -1.6567
8    8.4 0    -0.9632 -1.3577
9    8.3578  0.8412  -0.7938 -1.0573
10   8.2315  1.6739  -0.8866 -0.7805
11   9.4948  1.4172  -1.0506 -0.9654
12   9.5737  0.7105  -1.2571 -1.1636
13   9.6 0    -1.4902 -1.3854
14   9.5737  -0.7105  -1.742  -1.6094
15   9.4948  -1.4172 -2.5339 -1.8508
```

Table B-5: Actual Measurements of the 3D Printed Propeller.

Appendix C: MATLAB™ Scripting for RHINO™:
  OpenPVL_CADblade.txt

```
! Curve
0.551081,0.779187,0.970368
0.408407,0.854798,0.947187
0.307938,0.891982,0.871223
0.217712,0.915661,0.786151
0.134360,0.930011,0.695530
0.056403,0.937199,0.600818
-0.016790,0.938594,0.502699
-0.085518,0.935207,0.401566
-0.149205,0.927966,0.297235
-0.208660,0.917562,0.190392
-0.261703,0.905208,0.079957
-0.309369,0.891535,-0.033194
-0.352123,0.877104,-0.148589
-0.390713,0.862230,-0.265673
-0.425415,0.847278,-0.384213
-0.456820,0.832391,-0.505476
-0.483360,0.818753,-0.625879
-0.506512,0.806023,-0.741774
-0.528599,0.793120,-0.856066
-0.560211,0.773289,-0.967389
-0.541951,0.784945,-0.973347
-0.473148,0.824119,-0.871522
-0.396264,0.859941,-0.771845
-0.319915,0.888170,-0.674073
-0.244203,0.909611,-0.582370
-0.168731,0.924943,-0.494154
-0.093756,0.934495,-0.404805
-0.020636,0.938525,-0.315155
0.050020,0.937526,-0.224782
0.117458,0.932074,-0.133126
0.180721,0.922896,-0.039438
0.239116,0.910830,0.056967
0.294675,0.896017,0.155090
0.346629,0.879077,0.255596
0.395648,0.860197,0.358246
0.441581,0.839780,0.463350
0.484074,0.818372,0.571437
0.522135,0.796975,0.683684
0.553411,0.777696,0.803076
0.551081,0.779187,0.970368
Curve
0.561595,0.850421,0.957941
0.421788,0.927739,0.934312
0.321968,0.966924,0.859339
0.231925,0.992379,0.775449
0.148453,1.008249,0.686115
0.070141,1.016703,0.592759
-0.003612,1.019113,0.496051
-0.073089,1.016495,0.396375
-0.137742,1.009768,0.293545
-0.198340,0.999633,0.188234
-0.252801,0.987267,0.079372
-0.302112,0.973310,-0.032182
-0.346706,0.958332,-0.145974
-0.387294,0.942660,-0.261459
-0.424132,0.926670,-0.378415
-0.457792,0.910512,-0.498057
-0.486688,0.895399,-0.616984
-0.512225,0.881039,-0.731654
-0.536708,0.866343,-0.844802
-0.570325,0.844591,-0.954960
-0.552865,0.856122,-0.960923
-0.483651,0.897043,-0.860381
-0.406729,0.934439,-0.761993
-0.330337,0.964096,-0.665434
-0.254591,0.986807,-0.574723
-0.179041,1.003269,-0.487405
-0.103907,1.013809,-0.399020
-0.030516,1.018663,-0.310341
0.040547,1.018313,-0.220952

0.108566,1.013320,-0.130306
0.172621,1.004394,-0.037663
0.232063,0.992346,0.057636
0.288827,0.977335,0.154610
0.342184,0.959955,0.253907
0.392778,0.940388,0.355286
0.440468,0.919017,0.459046
0.484926,0.896355,0.565697
0.525225,0.873352,0.676378
0.559159,0.852025,0.793977
0.561595,0.850421,0.957941
Curve
0.581413,0.910851,0.933857
0.446833,0.983886,0.909289
0.348179,1.022967,0.836183
0.258442,1.049237,0.754535
0.174718,1.066379,0.667651
0.095722,1.076349,0.576889
0.020907,1.080395,0.482887
-0.049984,1.079441,0.386009
-0.116453,1.074304,0.286072
-0.179194,1.065636,0.183716
-0.236307,1.054443,0.077902
-0.288695,1.041319,-0.030549
-0.336727,1.026794,-0.141216
-0.381048,1.011184,-0.253579
-0.421876,0.994842,-0.367434
-0.459743,0.977920,-0.483905
-0.493042,0.961561,-0.599929
-0.523044,0.945577,-0.712166
-0.552018,0.928960,-0.823036
-0.589395,0.905706,-0.930891
-0.573431,0.915897,-0.936823
-0.503443,0.956157,-0.838714
-0.426444,0.992893,-0.742753
-0.349961,1.022359,-0.648478
-0.274125,1.045249,-0.559640
-0.198398,1.062228,-0.474025
-0.122934,1.073582,-0.387474
-0.049004,1.079486,-0.300652
0.022853,1.080356,-0.213153
0.091989,1.076675,-0.124454
0.157557,1.069049,-0.033841
0.218986,1.058176,0.059304
0.278031,1.044217,0.154032
0.334039,1.027672,0.250956
0.387598,1.008691,0.349837
0.438594,0.987586,0.450952
0.486744,0.964765,0.554773
0.531237,0.940998,0.662363
0.570143,0.917948,0.776421
0.581413,0.910851,0.933857
Curve
0.608651,1.001095,0.900030
0.480790,1.068405,0.873992
0.383599,1.107023,0.803361
0.294197,1.134061,0.724739
0.210069,1.152614,0.641190
0.130097,1.164355,0.553978
0.053806,1.170364,0.463697
-0.019029,1.171446,0.370684
-0.087974,1.168293,0.274762
-0.153626,1.161484,0.176524
-0.214327,1.151829,0.074977
-0.270864,1.139860,-0.029105
-0.323531,1.126044,-0.135353
-0.372831,1.110678,-0.243284
-0.419093,1.094079,-0.352716
-0.462637,1.076389,-0.464656
-0.501906,1.058649,-0.576487
-0.537997,1.040772,-0.685145

-0.573106,1.021859,-0.792637
-0.615645,0.996809,-0.897127
-0.601657,1.005314,-0.902934
-0.530466,1.044631,-0.808200
-0.453179,1.080406,-0.715568
-0.376384,1.109496,-0.624432
-0.300227,1.132480,-0.538188
-0.224074,1.149973,-0.454939
-0.148001,1.162215,-0.370933
-0.073193,1.169312,-0.286693
-0.000131,1.171600,-0.201833
0.070626,1.169470,-0.115866
0.138330,1.163405,-0.028112
0.202507,1.153966,0.061985
0.264681,1.141311,0.153527
0.324290,1.125825,0.247080
0.381857,1.107625,0.342411
0.437293,1.086932,0.439762
0.490376,1.064039,0.539557
0.540443,1.039504,0.642751
0.585912,1.014571,0.751780
0.608651,1.001095,0.900030
Curve
0.641804,1.119811,0.860044
0.521186,1.180785,0.831856
0.425496,1.218541,0.763840
0.336318,1.246105,0.688540
0.251574,1.265938,0.608721
0.170334,1.279404,0.525528
0.092201,1.287398,0.439501
0.016992,1.290581,0.350941
-0.054933,1.289524,0.259682
-0.124065,1.284717,0.166259
-0.189010,1.276779,0.069748
-0.250429,1.266165,-0.029146
-0.308541,1.253272,-0.130105
-0.363803,1.238360,-0.232694
-0.416344,1.221698,-0.336759
-0.466584,1.203407,-0.443168
-0.512963,1.184381,-0.549789
-0.556376,1.164618,-0.653887
-0.598877,1.143344,-0.757038
-0.647703,1.116409,-0.857267
-0.635906,1.123171,-0.862821
-0.562827,1.161514,-0.772129
-0.484646,1.196247,-0.683446
-0.406928,1.224866,-0.596045
-0.329848,1.247834,-0.512905
-0.252710,1.265712,-0.432476
-0.175503,1.278705,-0.351473
-0.099296,1.286868,-0.270278
-0.024500,1.290461,-0.188528
0.048425,1.289784,-0.105783
0.118841,1.285210,-0.021415
0.186379,1.277165,0.065066
0.252304,1.265793,0.152828
0.316162,1.251371,0.242382
0.378392,1.233981,0.333502
0.438936,1.213764,0.426396
0.497631,1.190904,0.521429
0.553961,1.165768,0.619436
0.606705,1.139209,0.722544
0.641804,1.119811,0.860044
Curve
0.678170,1.265770,0.815817
0.564253,1.320495,0.785121
0.469745,1.356992,0.719793
0.380492,1.384671,0.647983
0.294839,1.405404,0.572126
0.212041,1.420256,0.493234
0.131779,1.429938,0.411784
```

```
0.053907,1.434985,0.328041
-0.021284,1.435840,0.241854
-0.094183,1.432906,0.153692
-0.163660,1.426641,0.062717
-0.230250,1.417418,-0.030441
-0.294103,1.405558,-0.125524
-0.355580,1.391277,-0.222148
-0.414770,1.374793,-0.320191
-0.472032,1.356199,-0.420382
-0.525898,1.336233,-0.521070
-0.577069,1.314945,-0.619870
-0.627416,1.291680,-0.717937
-0.682975,1.263184,-0.813239
-0.673365,1.268333,-0.818394
-0.597971,1.305572,-0.732119
-0.518461,1.339137,-0.647699
-0.439372,1.367129,-0.564347
-0.360895,1.389908,-0.484614
-0.282309,1.407974,-0.407299
-0.203530,1.421501,-0.329597
-0.125498,1.430503,-0.251757
-0.048553,1.435176,-0.173448
0.026938,1.435745,-0.094280
0.100452,1.432480,-0.013686
0.171715,1.425694,0.068748
0.241742,1.415503,0.152278
0.310177,1.402098,0.237348
0.377382,1.385522,0.323748
0.443322,1.365853,0.411648
0.507886,1.343183,0.501353
0.570703,1.317720,0.593565
0.630883,1.289991,0.690083
0.678170,1.265770,0.815817
Curve
0.710457,1.439441,0.764090
0.600831,1.488536,0.734224
0.506600,1.523186,0.673090
0.416699,1.550194,0.605993
0.329784,1.570981,0.535154
0.245247,1.586377,0.461498
0.162827,1.596943,0.385461
0.082407,1.603106,0.307283
0.004231,1.605217,0.226821
-0.072017,1.603606,0.144503
-0.145390,1.598624,0.059542
-0.216332,1.590578,-0.027483
-0.284949,1.579729,-0.116347
-0.351540,1.566256,-0.206699
-0.416168,1.550336,-0.298432
-0.479169,1.532036,-0.392179
-0.539148,1.511972,-0.486604
-0.596636,1.490223,-0.579562
-0.653360,1.466240,-0.671930
-0.714519,1.437429,-0.761611
-0.706395,1.441438,-0.766569
-0.628389,1.477114,-0.685715
-0.546912,1.509180,-0.606651
-0.465845,1.536140,-0.528506
-0.385434,1.558261,-0.453534
-0.304960,1.575988,-0.380753
-0.224247,1.589482,-0.307693
-0.144141,1.598738,-0.234512
-0.064927,1.603909,-0.160899
0.013093,1.605169,-0.086496
0.089482,1.602726,-0.010776
0.164027,1.596820,0.066630
0.237574,1.587544,0.145027
0.309835,1.575037,0.224818
0.381117,1.559323,0.305800
0.451400,1.540447,0.388123
0.520609,1.518455,0.472051
0.588462,1.493470,0.558206
0.654289,1.465825,0.648181
0.710457,1.439441,0.764090
Curve
0.735751,1.638048,0.707402
0.629829,1.681621,0.679556
0.536032,1.713827,0.623236
0.445792,1.739483,0.561403
0.358025,1.759645,0.496094
0.272233,1.774943,0.428162
0.188209,1.785808,0.358008
0.105857,1.792575,0.285850
0.025387,1.795519,0.211551
-0.053460,1.794902,0.135504
-0.129887,1.790995,0.056967
-0.204256,1.784044,-0.023525
-0.276650,1.774260,-0.105771
-0.347312,1.761791,-0.189445
-0.416294,1.746777,-0.274452
-0.483917,1.729265,-0.361343
-0.548868,1.709759,-0.449016
-0.611525,1.688363,-0.535536
-0.673473,1.664622,-0.621575
-0.739220,1.636486,-0.705008
-0.732282,1.639602,-0.709795
-0.652076,1.673120,-0.635002
-0.568899,1.703199,-0.561954
-0.486126,1.728645,-0.489702
-0.404072,1.749645,-0.420242
-0.322038,1.766585,-0.352758
-0.239758,1.779620,-0.285060
-0.157995,1.788734,-0.217239
-0.076988,1.794047,-0.149004
0.003015,1.795696,-0.080025
0.081650,1.793841,-0.009809
0.158748,1.788668,0.061979
0.235023,1.780252,0.134682
0.310240,1.768695,0.208670
0.384661,1.754015,0.283750
0.458277,1.736236,0.360052
0.531037,1.715381,0.437811
0.602725,1.691525,0.517590
0.672836,1.664880,0.600822
0.735751,1.638048,0.707402
Curve
0.752770,1.857699,0.648862
0.650655,1.895879,0.622932
0.557858,1.925228,0.571419
0.467937,1.949036,0.514882
0.380031,1.968066,0.455165
0.293745,1.982781,0.393042
0.208910,1.993505,0.328875
0.125455,2.000492,0.262862
0.043556,2.003949,0.194873
-0.036997,2.004080,0.125265
-0.115537,2.001089,0.053349
-0.192361,1.995170,-0.020386
-0.267528,1.986488,-0.095764
-0.341235,1.975162,-0.172488
-0.413521,1.961303,-0.250476
-0.484687,1.944938,-0.330199
-0.553521,1.926479,-0.410770
-0.620273,1.906035,-0.490460
-0.686378,1.883240,-0.569767
-0.755694,1.856511,-0.646600
-0.749846,1.858881,-0.651124
-0.668276,1.889739,-0.582547
-0.584197,1.917400,-0.515630
-0.500506,1.940928,-0.449395
-0.417561,1.960446,-0.385592
-0.334696,1.976281,-0.323554
-0.251589,1.988570,-0.261360
-0.168914,1.997292,-0.199050
-0.086871,2.002539,-0.136354
-0.005661,2.004414,-0.072971
0.074417,2.003040,-0.008448
0.153234,1.998556,0.057512
0.231383,1.991022,0.124303
0.308676,1.980512,0.192260
0.385334,1.967035,0.261198
0.461356,1.950604,0.331233
0.536710,1.931230,0.402570
0.611237,1.908952,0.475710
0.684570,1.883898,0.551922
0.752770,1.857699,0.648862
Curve
0.760832,2.094175,0.590743
0.662764,2.127247,0.566203
0.571642,2.153523,0.519200
0.482798,2.175165,0.467719
0.395565,2.192707,0.413393
0.309632,2.206482,0.356910
0.224870,2.216725,0.298589
0.141224,2.223621,0.238607
0.058839,2.227324,0.176844
-0.022449,2.227988,0.113617
-0.102099,2.225761,0.048304
-0.180345,2.220791,-0.018660
-0.257225,2.213204,-0.087126
-0.332899,2.203092,-0.156830
-0.407389,2.190541,-0.227703
-0.480972,2.175569,-0.300146
-0.552550,2.158500,-0.373471
-0.622278,2.139440,-0.446162
-0.691430,2.118103,-0.518558
-0.763243,2.093298,-0.588674
-0.758422,2.095049,-0.592813
-0.676455,2.122933,-0.530323
-0.592421,2.147900,-0.469352
-0.508745,2.169243,-0.408955
-0.425792,2.187038,-0.350643
-0.342946,2.201550,-0.293893
-0.259861,2.212896,-0.237052
-0.177125,2.221050,-0.180115
-0.094900,2.226079,-0.122837
-0.013345,2.228061,-0.064951
0.067299,2.227085,-0.006051
0.146938,2.223251,0.054122
0.226051,2.216605,0.115021
0.304494,2.207197,0.176943
0.382447,2.195033,0.239718
0.459917,2.180117,0.303444
0.536888,2.162449,0.368295
0.613250,2.142046,0.434703
0.688751,2.118975,0.503762
0.760832,2.094175,0.590743
Curve
0.758333,2.343572,0.533562
0.664551,2.371871,0.510363
0.575839,2.394955,0.467709
0.488908,2.414201,0.421123
0.403251,2.429977,0.372028
0.318628,2.442514,0.321027
0.234939,2.451979,0.268400
0.152144,2.458506,0.214299
0.070360,2.462204,0.158620
-0.010541,2.463186,0.101638
-0.090121,2.461560,0.042800
-0.168565,2.457434,-0.017509
-0.245896,2.450905,-0.079167
-0.322239,2.442040,-0.141944
-0.397607,2.430907,-0.205785
```

-0.472245,2.417516,-0.271026
-0.545180,2.402119,-0.337147
-0.616496,2.384813,-0.402835
-0.687307,2.365377,-0.468301
-0.760298,2.342935,-0.531708
-0.756368,2.344207,-0.535417
-0.675057,2.368902,-0.478895
-0.592062,2.390996,-0.423728
-0.509390,2.409963,-0.369042
-0.427394,2.425847,-0.316128
-0.345515,2.438856,-0.264588
-0.263408,2.449084,-0.213011
-0.181577,2.456507,-0.161361
-0.100153,2.461172,-0.109420
-0.019261,2.463134,-0.056955
0.060907,2.462456,-0.003605
0.140284,2.459211,0.050849
0.219255,2.453431,0.105925
0.297707,2.445152,0.161882
0.375791,2.434374,0.218565
0.453516,2.421099,0.276057
0.530876,2.405321,0.334504
0.607801,2.387043,0.394269
0.684127,2.366298,0.456283
0.758333,2.343572,0.533562
Curve
0.738257,2.603385,0.473866
0.649941,2.626826,0.452099
0.565210,2.646351,0.413929
0.481847,2.662792,0.372394
0.399471,2.676389,0.328704
0.317900,2.687299,0.283373
0.237061,2.695633,0.236640
0.156922,2.701483,0.188636
0.077577,2.704925,0.139269
-0.001070,2.706037,0.088774
-0.078675,2.704893,0.036675
-0.155382,2.701572,-0.016698
-0.231200,2.696142,-0.071250
-0.306223,2.688655,-0.126787
-0.380461,2.679158,-0.183265
-0.454118,2.667660,-0.240966
-0.526357,2.654352,-0.299518
-0.597217,2.639312,-0.357805
-0.667645,2.622382,-0.415934
-0.739819,2.602941,-0.472261
-0.736696,2.603827,-0.475471
-0.657875,2.624850,-0.425137
-0.577720,2.643648,-0.375966
-0.497845,2.659847,-0.327188
-0.418568,2.673469,-0.279892
-0.339398,2.684669,-0.233790
-0.260012,2.693516,-0.187704
-0.180836,2.699988,-0.141574
-0.101971,2.704115,-0.095212
-0.023514,2.705935,-0.048420
0.054385,2.705490,-0.000886
0.131681,2.702831,0.047570
0.208675,2.697979,0.096536
0.285283,2.690957,0.146230
0.361625,2.681765,0.196519
0.437712,2.670402,0.247467
0.513548,2.656860,0.299193
0.589090,2.641138,0.351993
0.664246,2.623245,0.406630
0.738257,2.603385,0.473866
Curve
0.717675,2.864212,0.423219
0.633930,2.883904,0.403005
0.552740,2.900560,0.368727
0.472617,2.914687,0.331528
0.393273,2.926449,0.292453
0.314568,2.935952,0.251946
0.236445,2.943274,0.210215
0.158883,2.948478,0.167372
0.081955,2.951618,0.123339
0.005590,2.952751,0.078317
-0.069938,2.951928,0.031890
-0.144739,2.949206,-0.015653
-0.218819,2.944637,-0.064237
-0.292248,2.938258,-0.113695
-0.365032,2.930106,-0.163992
-0.437350,2.920187,-0.215368
-0.508470,2.908647,-0.267551
-0.578401,2.895552,-0.319579
-0.647960,2.880784,-0.371492
-0.718944,2.863894,-0.421811
-0.716406,2.864530,-0.424626
-0.639992,2.882564,-0.379587
-0.562494,2.898684,-0.335562
-0.485243,2.912612,-0.291864
-0.408531,2.924358,-0.249425
-0.331922,2.934041,-0.208035
-0.255117,2.941714,-0.166691
-0.178476,2.947357,-0.125321
-0.102083,2.950991,-0.083761
-0.026011,2.952641,-0.041839
0.049619,2.952339,0.000717
0.124777,2.950118,0.044058
0.199703,2.945995,0.087828
0.274334,2.939984,0.132212
0.348766,2.932086,0.177095
0.423013,2.922298,0.222527
0.497084,2.910614,0.268608
0.570953,2.897029,0.315584
0.644574,2.881543,0.364094
0.717675,2.864212,0.423219
Curve
0.714271,3.118727,0.389780
0.632399,3.136353,0.371044
0.552413,3.151425,0.339482
0.473304,3.164273,0.305242
0.394843,3.175018,0.269280
0.316918,3.183740,0.232003
0.239482,3.190500,0.193600
0.162520,3.195345,0.154173
0.086098,3.198316,0.113649
0.010154,3.199459,0.072212
-0.065074,3.198813,0.029478
-0.139677,3.196425,-0.014289
-0.213658,3.192333,-0.059020
-0.287077,3.186570,-0.104561
-0.359937,3.179164,-0.150881
-0.432402,3.170120,-0.198189
-0.503823,3.159557,-0.246258
-0.574154,3.147537,-0.294218
-0.644145,3.133962,-0.342084
-0.715380,3.118472,-0.388464
-0.713162,3.118981,-0.391095
-0.637159,3.135390,-0.349671
-0.560202,3.150050,-0.309202
-0.483480,3.162734,-0.269022
-0.407286,3.173446,-0.229964
-0.331222,3.182284,-0.191847
-0.254983,3.189298,-0.153774
-0.178895,3.194470,-0.115669
-0.103027,3.197816,-0.077382
-0.027441,3.199357,-0.038754
0.047763,3.199118,0.000465
0.122560,3.197127,0.040412
0.197159,3.193394,0.080758
0.271507,3.187934,0.121672
0.345689,3.180745,0.163047
0.419719,3.171825,0.204929
0.493605,3.161170,0.247407
0.567335,3.148773,0.290708
0.640882,3.134631,0.335414
0.714271,3.118727,0.389780
Curve
0.711613,3.367945,0.361948
0.631170,3.383944,0.344581
0.552107,3.397739,0.315311
0.473777,3.409543,0.283551
0.395996,3.419450,0.250188
0.318670,3.427521,0.215602
0.241764,3.433803,0.179967
0.165266,3.438333,0.143377
0.089235,3.441146,0.105762
0.013619,3.442276,0.067296
-0.061371,3.441756,0.027615
-0.135813,3.439623,-0.013035
-0.209707,3.435909,-0.054588
-0.283107,3.430641,-0.096897
-0.356015,3.423843,-0.139934
-0.428597,3.415517,-0.183863
-0.500230,3.405763,-0.228543
-0.570864,3.394638,-0.273117
-0.641180,3.382061,-0.317610
-0.712607,3.367735,-0.360698
-0.710620,3.368155,-0.363198
-0.634905,3.383245,-0.324835
-0.558326,3.396722,-0.287390
-0.481974,3.408394,-0.250204
-0.406151,3.418258,-0.214031
-0.330484,3.426402,-0.178704
-0.254665,3.432870,-0.143406
-0.178989,3.437646,-0.108066
-0.103517,3.440746,-0.072541
-0.028302,3.442187,-0.036684
0.046571,3.441988,-0.000263
0.121083,3.440173,0.036853
0.195418,3.436752,0.074351
0.269532,3.431735,0.112391
0.343500,3.425121,0.150870
0.417336,3.416911,0.189832
0.491050,3.407098,0.229361
0.564634,3.395679,0.269670
0.638078,3.382648,0.311305
0.711613,3.367945,0.361948
Curve
0.706412,3.608924,0.337282
0.627458,3.623485,0.321047
0.549485,3.636126,0.293757
0.472125,3.646978,0.264156
0.395231,3.656110,0.233067
0.318726,3.663572,0.200839
0.242583,3.669401,0.167637
0.166792,3.673626,0.133547
0.091407,3.676274,0.098501
0.016386,3.677374,0.062661
-0.058087,3.676952,0.025687
-0.132076,3.675038,-0.012195
-0.205582,3.671660,-0.050918
-0.278651,3.666838,-0.090346
-0.351284,3.660594,-0.130450
-0.423644,3.652927,-0.171393
-0.495151,3.643923,-0.212991
-0.565735,3.633633,-0.254515
-0.636023,3.621992,-0.295967
-0.707308,3.608748,-0.336966
-0.705517,3.609099,-0.338468
-0.630353,3.622983,-0.302836
-0.554403,3.635380,-0.268088

```
-0.478671,3.646124,-0.233572    -0.191297,4.105359,-0.045647    0.498505,4.441627,0.209402
-0.403463,3.655211,-0.199968    -0.261195,4.101505,-0.079205    0.437958,4.448005,0.190721
-0.328434,3.662715,-0.167120    -0.330772,4.096481,-0.113305    0.377662,4.453530,0.170735
-0.253272,3.668678,-0.134285    -0.400163,4.090286,-0.148068    0.317565,4.458218,0.149907
-0.178248,3.673088,-0.101394    -0.468906,4.082976,-0.183356    0.257641,4.462082,0.128435
-0.103415,3.675956,-0.068316    -0.536915,4.074591,-0.218599    0.197879,4.465132,0.106410
-0.028819,3.677298,-0.034912    -0.604688,4.065085,-0.253792    0.138278,4.467375,0.083880
0.045469,3.677129,-0.000968     -0.673211,4.054301,-0.287905    0.078866,4.468818,0.060805
0.119433,3.675471,0.033640      -0.671858,4.054525,-0.289855    0.019626,4.469471,0.037277
0.193236,3.672330,0.068618      -0.600383,4.065724,-0.259452    -0.039353,4.469341,0.013099
0.266842,3.667716,0.104113      -0.528305,4.075716,-0.229787    -0.098097,4.468438,-0.011602
0.340319,3.661630,0.140029      -0.456411,4.084392,-0.200298    -0.156605,4.466770,-0.036785
0.413678,3.654069,0.176410      -0.384969,4.091744,-0.171505    -0.214897,4.464345,-0.062373
0.486932,3.645030,0.213332      -0.313705,4.097823,-0.143285    -0.272975,4.461170,-0.088347
0.560076,3.634510,0.250999      -0.242341,4.102662,-0.115064    -0.330936,4.457246,-0.114789
0.633109,3.622502,0.289924      -0.171092,4.106251,-0.086784    -0.388452,4.452602,-0.141611
0.706412,3.608924,0.337282      -0.099991,4.108597,-0.058340    -0.445452,4.447261,-0.168414
Curve                           -0.029074,4.109711,-0.029619    -0.502285,4.441201,-0.195186
0.696795,3.838356,0.314484      0.041612,4.109603,-0.000450     -0.559623,4.434341,-0.221184
0.619673,3.851559,0.299011      0.112057,4.108286,0.029266      -0.558709,4.434456,-0.222583
0.543213,3.863085,0.273427      0.182383,4.105765,0.059290      -0.499365,4.441530,-0.199255
0.467260,3.873005,0.245729      0.252564,4.102046,0.089740      -0.439612,4.447842,-0.176459
0.391699,3.881375,0.216670      0.322652,4.097129,0.120539      -0.379990,4.453332,-0.153783
0.316469,3.888232,0.186571      0.392657,4.091013,0.151721      -0.320702,4.457994,-0.131586
0.241546,3.893605,0.155579      0.462591,4.083696,0.183349      -0.261552,4.461855,-0.109786
0.166925,3.897517,0.123775      0.532455,4.075176,0.215587      -0.202332,4.464932,-0.087985
0.092655,3.899989,0.091093      0.602263,4.065445,0.248851      -0.143192,4.467220,-0.066139
0.018698,3.901045,0.057684      0.672534,4.054413,0.288880      -0.084154,4.468722,-0.044173
-0.054781,3.900705,0.023232     Curve                           -0.025239,4.469443,-0.022008
-0.127838,3.898995,-0.012055    0.628739,4.254078,0.258738      0.033520,4.469389,0.000478
-0.200472,3.895936,-0.048115    0.560231,4.263641,0.244431      0.092121,4.468565,0.023355
-0.272723,3.891545,-0.084820    0.491962,4.272056,0.222751      0.150643,4.466975,0.046452
-0.344593,3.885841,-0.122143    0.424012,4.279335,0.199521      0.209071,4.464622,0.069853
-0.416236,3.878821,-0.160226    0.356315,4.285502,0.175289      0.267440,4.461506,0.093502
-0.487121,3.870557,-0.198900    0.288833,4.290579,0.150290      0.325759,4.457627,0.117423
-0.557165,3.861097,-0.237510    0.221554,4.294578,0.124635      0.384035,4.452985,0.141661
-0.626937,3.850383,-0.276057    0.154475,4.297514,0.098379      0.442273,4.447578,0.166332
-0.697593,3.838211,-0.313378    0.087630,4.299397,0.071476      0.500491,4.441404,0.191728
-0.695998,3.838501,-0.315589    0.020996,4.300238,0.044036      0.559166,4.434399,0.221884
-0.621871,3.851205,-0.282469    -0.045316,4.300051,0.015824     Curve
-0.547037,3.862545,-0.250183    -0.111341,4.298848,-0.013006    0.413872,4.596223,0.159362
-0.472410,3.872381,-0.218104    -0.177078,4.296642,-0.042409    0.369029,4.600040,0.150646
-0.398287,3.880705,-0.186835    -0.242550,4.293444,-0.072293    0.324239,4.603414,0.137367
-0.324356,3.887582,-0.156230    -0.307761,4.289263,-0.102637    0.279629,4.606339,0.123119
-0.250309,3.893051,-0.125625    -0.372823,4.284098,-0.133541    0.235162,4.608823,0.108243
-0.176394,3.897100,-0.094955    -0.437347,4.277992,-0.164900    0.190820,4.610872,0.092887
-0.102654,3.899739,-0.064100    -0.501256,4.270976,-0.196235    0.146595,4.612490,0.077119
-0.029128,3.900981,-0.032932    -0.564967,4.263016,-0.227532    0.102487,4.613681,0.060974
0.044122,3.900840,-0.001256     -0.629291,4.253996,-0.257916    0.058515,4.614448,0.044423
0.117083,3.899333,0.031045      -0.628186,4.254159,-0.259561    0.014667,4.614795,0.027535
0.189902,3.896465,0.063694      -0.561444,4.263481,-0.232314    -0.028992,4.614728,0.010162
0.262546,3.892245,0.096831      -0.494209,4.271797,-0.205688    -0.072481,4.614250,-0.007598
0.335076,3.886673,0.130365      -0.427129,4.279025,-0.179208    -0.115799,4.613366,-0.025720
0.407504,3.879748,0.164336      -0.360434,4.285158,-0.153307    -0.158960,4.612080,-0.044147
0.479840,3.871467,0.198817      -0.293892,4.290235,-0.127890    -0.201963,4.610397,-0.062867
0.552084,3.861827,0.233996      -0.227265,4.294280,-0.102479    -0.244880,4.608317,-0.081943
0.624244,3.850821,0.270351      -0.160730,4.297285,-0.077019    -0.287466,4.605857,-0.101320
0.696795,3.838356,0.314484      -0.094318,4.299255,-0.051423    -0.329671,4.603028,-0.120701
Curve                           -0.028053,4.300198,-0.025597    -0.371752,4.599821,-0.140062
0.672534,4.054413,0.288880      0.038025,4.300121,0.000604      -0.414200,4.596193,-0.158849
0.598742,4.065965,0.273860      0.103910,4.299034,0.027264      -0.413543,4.596252,-0.159876
0.525362,4.076096,0.250036      0.169700,4.296940,0.054178      -0.369653,4.599990,-0.143052
0.452390,4.084839,0.224366      0.235375,4.293843,0.081450      -0.325471,4.603327,-0.126612
0.379737,4.092232,0.197507      0.300978,4.289744,0.109010      -0.281385,4.606232,-0.110261
0.307354,4.098305,0.169738      0.366519,4.284642,0.136889      -0.237541,4.608701,-0.094270
0.235225,4.103076,0.141189      0.432006,4.278535,0.165138      -0.193791,4.610748,-0.078588
0.163345,4.106566,0.111929      0.497445,4.271421,0.193894      -0.149986,4.612381,-0.062921
0.091756,4.108789,0.081901      0.562852,4.263295,0.223501      -0.106235,4.613596,-0.047229
0.020428,4.109763,0.051235      0.628739,4.254078,0.258738      -0.062554,4.614395,-0.031457
-0.050502,4.109503,0.019656     Curve                           -0.018960,4.614780,-0.015547
-0.121077,4.108030,-0.012657    0.559166,4.434399,0.221884      0.024527,4.614754,0.000593
```

```
0.067903,4.614319,0.017012        -0.307877,4.815082,-0.113960      0.013953,4.917356,0.005306
0.111226,4.613478,0.033585        -0.275279,4.817055,-0.102088      -0.014104,4.917355,-0.004883
0.154486,4.612232,0.050374        -0.242562,4.818813,-0.090384      -0.042159,4.917195,-0.015076
0.197707,4.610582,0.067337        -0.209885,4.820347,-0.078711      -0.070213,4.916874,-0.025273
0.240896,4.608527,0.084491        -0.177316,4.821655,-0.067161      -0.098264,4.916394,-0.035473
0.284061,4.606068,0.101867        -0.144800,4.822741,-0.055701      -0.126314,4.915753,-0.045676
0.327205,4.603204,0.119548        -0.112270,4.823608,-0.044225      -0.154364,4.914952,-0.055882
0.370346,4.599934,0.137741        -0.079767,4.824255,-0.032723      -0.182410,4.913991,-0.066089
0.413872,4.596223,0.159362        -0.047297,4.824683,-0.021175      -0.210451,4.912870,-0.076293
Curve                             -0.014865,4.824892,-0.009559      -0.238491,4.911589,-0.086497
0.345491,4.721287,0.129890        0.017520,4.824883,0.002161        -0.266536,4.910147,-0.096694
0.308256,4.723865,0.122178        0.049858,4.824657,0.014006        -0.266526,4.910147,-0.096709
0.271017,4.726147,0.111069        0.082172,4.824215,0.025924        -0.238461,4.911590,-0.086543
0.233900,4.728130,0.099245        0.114457,4.823557,0.037941        -0.210391,4.912873,-0.076383
0.196883,4.729816,0.086960        0.146723,4.822683,0.050040        -0.182322,4.913994,-0.066224
0.159953,4.731208,0.074322        0.178973,4.821594,0.062230        -0.154258,4.914955,-0.056068
0.123106,4.732311,0.061380        0.211208,4.820289,0.074524        -0.126197,4.915756,-0.045912
0.086342,4.733124,0.048162        0.243429,4.818770,0.086961        -0.098136,4.916396,-0.035755
0.049676,4.733651,0.034645        0.275643,4.817034,0.099631        -0.070077,4.916876,-0.025596
0.013098,4.733893,0.020878        0.308007,4.815073,0.113751        -0.042019,4.917196,-0.015434
-0.023343,4.733854,0.006754       Curve                             -0.013962,4.917356,-0.005268
-0.059661,4.733536,-0.007657      0.280976,4.878307,0.102547        0.014092,4.917355,0.004902
-0.095855,4.732941,-0.022334      0.251285,4.879927,0.092762        0.042145,4.917195,0.015077
-0.131935,4.732073,-0.037233      0.221574,4.881366,0.082382        0.070196,4.916874,0.025256
-0.167903,4.730933,-0.052344      0.191880,4.882623,0.071883        0.098246,4.916394,0.035439
-0.203813,4.729522,-0.067708      0.162200,4.883699,0.061308        0.126295,4.915753,0.045627
-0.239484,4.727850,-0.083279      0.132532,4.884595,0.050675        0.154343,4.914953,0.055818
-0.274873,4.725925,-0.098840      0.102877,4.885309,0.039993        0.182390,4.913992,0.066014
-0.310169,4.723739,-0.114385      0.073235,4.885843,0.029265        0.210436,4.912871,0.076217
-0.345736,4.721269,-0.129501      0.043608,4.886198,0.018487        0.238481,4.911589,0.086430
-0.345247,4.721305,-0.130279      0.013995,4.886372,0.007669        0.266531,4.910147,0.096702
-0.308604,4.723842,-0.116652      -0.015596,4.886367,-0.003210      SelNone
-0.271740,4.726106,-0.103323      -0.045165,4.886183,-0.014138      SelLast
-0.234951,4.728077,-0.090059      -0.074715,4.885821,-0.025109      -Patch
-0.198351,4.729754,-0.077050      -0.104247,4.885280,-0.036114      USpans=10
-0.161833,4.731144,-0.064250      -0.133761,4.884561,-0.047150      VSpans=10
-0.125279,4.732254,-0.051447      -0.163269,4.883664,-0.058223      Stiffness=0.0001
-0.088770,4.733079,-0.038615      -0.192739,4.882590,-0.069316      AutomaticTrim=Yes
-0.052317,4.733622,-0.025713      -0.222162,4.881339,-0.080397      enter
-0.015930,4.733885,-0.012697      -0.251568,4.879912,-0.091472      SelNone
0.020373,4.733868,0.000498        -0.281021,4.878305,-0.102474      SelCrv
0.056592,4.733573,0.013913        -0.280932,4.878310,-0.102619      -Loft Tight RebuildCount=10
0.092770,4.733002,0.027451        -0.251282,4.879927,-0.091895      enter
0.128898,4.732156,0.041160        -0.221589,4.881365,-0.081233      enter
0.164995,4.731035,0.055009        -0.191912,4.882622,-0.070580      SelNone
0.201065,4.729640,0.069010        -0.162274,4.883697,-0.059963      SelCrv
0.237115,4.727969,0.083189        -0.132658,4.884591,-0.049367      Delete
0.273146,4.726025,0.097611        -0.103039,4.885306,-0.038759      SelAll
0.309172,4.723805,0.112437        -0.073432,4.885840,-0.028138      Join
0.345491,4.721287,0.129890        -0.043837,4.886196,-0.017496      Dir
Curve                             -0.014257,4.886371,-0.006827      SelLast
0.308007,4.815073,0.113751        0.015306,4.886368,0.003880        Flip
0.275182,4.817061,0.104857        0.044852,4.886186,0.014634        enter
0.242329,4.818825,0.094183        0.074388,4.885826,0.025417        SelLast
0.209534,4.820362,0.083143        0.103914,4.885287,0.036236        Rotate
0.176786,4.821675,0.071867        0.133432,4.884570,0.047088        Copy=Yes
0.144080,4.822763,0.060411        0.162942,4.883675,0.057975        0,0,0
0.111415,4.823628,0.048800        0.192447,4.882601,0.068903        120.000000
0.078791,4.824271,0.037049        0.221944,4.881349,0.079885        240.000000
0.046215,4.824693,0.025143        0.251437,4.879912,0.090956        enter
0.013683,4.824895,0.013111        0.280976,4.878307,0.102547        SelNone
-0.018779,4.824878,0.000894       Curve                             Circle
-0.051178,4.824643,-0.011472      0.266531,4.910147,0.096702        0,0,0
-0.083514,4.824192,-0.023972      0.238466,4.911590,0.086623        0.988136
-0.115794,4.823525,-0.036583      0.210397,4.912872,0.076483        SelNone
-0.148017,4.822643,-0.049298      0.182330,4.913994,0.066332        SelLast
-0.180215,4.821548,-0.062136      0.154264,4.914955,0.056173        ExtrudeCrv BothSides=Yes
-0.212295,4.820242,-0.075062      0.126200,4.915756,0.046009        Cap=Yes DeleteInput=Yes
-0.244227,4.818729,-0.087970      0.098136,4.916396,0.035840         2.425920
-0.276112,4.817008,-0.100867      0.070074,4.916876,0.025666        SelNone
-0.308137,4.815065,-0.113541      0.042013,4.917196,0.015488        BooleanUnion
```

```
SelAll
enter
-Mesh
SelAll
enter
enter
Weld
SelLast
enter
180
enter
UnifyMeshNormals
SelLast
enter
SelNakedMeshEdgePt
SelLast
enter
Zoom All Extents
enter
```